

# **MAINVIEW<sup>®</sup> SRM**

## **User Guide and Reference**

**Version 7.2**

**June 20, 2003**



Copyright 2003 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc. IBM and DB2 are registered trademarks of International Business Machines Corp. All other registered trademarks or trademarks belong to their respective companies.

THE USE AND CONTENTS OF THIS DOCUMENTATION ARE GOVERNED BY THE SOFTWARE LICENSE AGREEMENT ENCLOSED AT THE BACK OF THIS DOCUMENTATION.

## **Restricted Rights Legend**

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

---

## **Contacting BMC Software**

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

### **United States and Canada**

**Address** BMC Software, Inc.  
2101 CityWest Blvd.  
Houston TX 77042-2827

**Telephone** 713 918 8800 or  
800 841 2031

**Fax** 713 918 8000

### **Outside United States and Canada**

**Telephone** (01) 713 918 8800

**Fax** (01) 713 918 8000

---

---

## Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

### Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at <http://www.bmc.com/support.html>. From this Web site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

### Support by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at [www.bmc.com/support.html](http://www.bmc.com/support.html).

### Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as `file system full`
  - messages from related software

---

---

# Contents

<b>About This Book</b> .....	<b>xiii</b>
<b>Chapter 1</b>	<b>Introducing MAINVIEW SRM</b>
	Overview..... 1-2
	MAINVIEW SRM Components..... 1-2
	Accessing MAINVIEW SRM ..... 1-3
	Using Easy Menus ..... 1-4
	Using EZCmd Menus ..... 1-5
<b>Chapter 2</b>	<b>Using MAINVIEW SRM Operator Services</b>
	Overview..... 2-2
	Activating System Software in OS/390 ..... 2-2
	Starting and Stopping MAINVIEW SRM ..... 2-3
	Preparing to Start SVOS..... 2-4
	Parameter Descriptions ..... 2-5
	Starting SVOS ..... 2-8
	Starting MAINVIEW SRM Components and Subcomponents..... 2-8
	Viewing the Status of Components and Subcomponents..... 2-10
	Stopping MAINVIEW SRM Components..... 2-12
	Stopping SVOS ..... 2-13
	Starting and Stopping MAINVIEW SRM with Other Products..... 2-13
	Communicating with MAINVIEW SRM Components ..... 2-14
	Common Command Structure ..... 2-14
	Controlling MAINVIEW SRM Components..... 2-14
	SVOS Command Syntax Conventions..... 2-15
	SVOS Loader Console Commands ..... 2-16
	SVOS Component Console Commands..... 2-21
	Tracking MAINVIEW SRM Activity..... 2-25
	Audit Log Views ..... 2-26
	Audit Log Records..... 2-26
<b>Chapter 3</b>	<b>Defining the MAINVIEW SRM System</b>
	Overview..... 3-2
	Defining the System Configuration ..... 3-3

	MAINVIEW SRM Parameters . . . . .	3-6
	Action Parameters . . . . .	3-6
	Selection Parameters . . . . .	3-6
	Defining the Sysplex Environment . . . . .	3-10
	Sysplex Parameters . . . . .	3-11
	Syntax Considerations for the Shared Parmlib Environment . . . . .	3-11
	Ordering of Selection Criteria . . . . .	3-12
	Editing Definition Members . . . . .	3-13
<b>Chapter 4</b>	<b>Defining the SMFUNCxx Member and Functions</b>	
	Overview . . . . .	4-2
	Defining the SMFUNCxx Member . . . . .	4-2
	Active and Inactive Functions . . . . .	4-2
	SMFUNCxx Member Parameter Reference . . . . .	4-2
	Defining and Activating Functions . . . . .	4-4
	Defining Parameters for Functions . . . . .	4-4
	Function Parameter Reference . . . . .	4-5
	Defining the USERVARS Function . . . . .	4-5
<b>Chapter 5</b>	<b>Using Filter and Rule List Parameters</b>	
	Overview . . . . .	5-2
	Using SMFLSTxx—Filter List Parameters . . . . .	5-2
	SET Statement Parameters . . . . .	5-2
	INC and EXC Parameters . . . . .	5-6
	Using SMRLSTxx—Rule List Parameters . . . . .	5-7
	Defining Filter and Rule Lists . . . . .	5-10
<b>Chapter 6</b>	<b>Defining a Pool</b>	
	Overview . . . . .	6-2
	Using the SMPOOLxx Member . . . . .	6-2
	Parameter Quick Reference . . . . .	6-2
	Defining Pools . . . . .	6-5
<b>Chapter 7</b>	<b>Using Variables</b>	
	Overview . . . . .	7-2
	Using the SMVARSxx Member . . . . .	7-2
	Usage Notes . . . . .	7-3
	Examples . . . . .	7-4
	Defining the Variables Option . . . . .	7-5
<b>Chapter 8</b>	<b>Defining a Calendar</b>	
	Overview . . . . .	8-2
	Using the SMCALSxx Member . . . . .	8-2
	Defining the Calendars Option . . . . .	8-3
<b>Chapter 9</b>	<b>Using Diagnostics</b>	
	Passive Diagnostics . . . . .	9-2
	Active Diagnostics . . . . .	9-2

---

	Using the SMDIAGxx Member .....	9-3
	Parameter Quick Reference .....	9-4
	Defining the Diagnostics Option .....	9-5
	Using the Tracing Facility .....	9-5
	Activating Tracing .....	9-5
	How Function Tracing Works .....	9-7
	Additional Trace Options .....	9-10
<b>Appendix A</b>	<b>MAINVIEW SRM General Syntax Rules</b>	
	Statements and Parameters .....	A-2
	Parameter Relationships .....	A-4
	Comparison Operators .....	A-6
	Name Masking .....	A-6
	Variables in Name Masks .....	A-8
	Field-to-Field Comparisons .....	A-10
	Selection Criteria Ordering .....	A-11
<b>Appendix B</b>	<b>Processing SMF Records</b>	
	Overview .....	B-2
	Using the PRSMF000 Utility .....	B-2
	SMF Record Layout .....	B-3
<b>Appendix C</b>	<b>Altering Messages with the Message Exit</b>	
<b>Appendix D</b>	<b>Global Parameters</b>	
	Master System Member Parameters .....	D-2
	Subordinate Members .....	D-2
	Other Parmlib Members .....	D-2
	Parameter Quick Reference .....	D-2
	Usage Notes .....	D-22
	Parameter Explanations .....	D-23
	Pool Member Parameters .....	D-99
	Parameter Quick Reference .....	D-99
	Parameter Explanations .....	D-100
	SMS Subpool Member Parameters .....	D-103
	Parameter Quick Reference .....	D-103
	Parameter Explanations .....	D-103
	Calendar Member Parameters .....	D-104
	Parameter Quick Reference .....	D-104
	Parameter Explanations .....	D-105
	Variable Member Parameters .....	D-106
	Parameter Quick Reference .....	D-106
	Parameter Explanations .....	D-107
	Function Member Parameters .....	D-107
	Subordinate Members .....	D-108
	Parameter Quick Reference .....	D-108
	Parameter Explanations .....	D-108
	Diagnostic Member Parameters .....	D-112

---

	Parameter Quick Reference .....	D-112
	Event Member Parameters .....	D-113
	Parameter Quick Reference .....	D-113
	Parameter Explanations .....	D-114
	VTOC Scan Facility Parameters .....	D-116
	SVOS VTOC Command.....	D-116
	Parameter Quick Reference .....	D-116
	Parameter Explanations .....	D-118
<b>Appendix E</b>	<b>Filter and Rule List Parameters</b>	
	Parameter Explanations.....	E-18
<b>Appendix F</b>	<b>Troubleshooting</b>	
<b>Glossary</b>		
<b>Index</b>		



---

# List of Figures

Figure 1-1	Storage Management Solutions Submenu .....	1-3
Figure 1-2	EZSRM Menu .....	1-4
Figure 2-1	Component Status (COMPSTAT) .....	2-11
Figure 2-2	Component Summary (COMPSUMM) .....	2-11
Figure 3-1	MAINVIEW SRM Partitioned Data Set Members .....	3-2
Figure 3-2	Connecting Definition Member Parameters .....	3-5
Figure 6-1	Example Storage Configuration for Pools .....	6-4
Figure 9-1	Tracing Example .....	9-7
Figure 9-2	Tracing a Function .....	9-9
Figure B-1	SMF Record Layout (Part 1 of 7) .....	B-3



---

---

# List of Tables

Table 1-1	New Product Names .....	1-2
Table 2-1	Execution JCL DD Statements and Start-Up Options .....	2-4
Table 2-2	Startup Override Keywords .....	2-9
Table 2-3	Component Start Commands .....	2-9
Table 2-4	Shutdown Commands .....	2-12
Table 2-5	MAINVIEW SRM Components and Subcomponents .....	2-14
Table 2-6	SVOS Loader Commands .....	2-16
Table 2-7	DISPLAY Command Syntax .....	2-22
Table 2-8	REFRESH Command Syntax .....	2-23
Table 2-9	SET Command Syntax .....	2-24
Table 2-10	Audit Log Views .....	2-26
Table 2-11	Audit Log Records .....	2-26
Table 3-1	MAINVIEW SRM Definition Members .....	3-4
Table 4-1	SET Statement Function Parameters .....	4-2
Table 4-2	SET Statement Parameters for USERVARS Function .....	4-6
Table 5-1	SET Statement Parameter Quick Reference .....	5-2
Table 6-1	SET Statement Pool Parameter Quick Reference .....	6-3
Table 6-2	INC and EXC Statement Pool Parameter Quick Reference .....	6-3
Table 7-1	SET Statement Variable Parameter .....	7-2
Table 7-2	INC and EXC Statement Variable Parameter .....	7-2
Table 8-1	SET Statement Calendar Parameter .....	8-2
Table 8-2	INC and EXC Statement Calendar Parameters .....	8-2
Table 9-1	SMDIAGxx SET and INC Statements .....	9-3
Table 9-2	SET Statement Diagnostics Parameters .....	9-4
Table 9-3	INC and EXC Statement Diagnostic Parameters .....	9-4
Table D-1	SET Statement System Parameters .....	D-3
Table D-2	SET Statement Pool Parameter Quick Reference .....	D-99
Table D-3	INC and EXC Statement Pool Parameter Quick Reference .....	D-99
Table D-4	SET Statement SMS Pool Parameters .....	D-103
Table D-5	INC and EXC Statement SMS Pool Parameters .....	D-103
Table D-6	SET Statement Calendar Parameters .....	D-104
Table D-7	INC and EXC Statement Calendar Parameters .....	D-105
Table D-8	SET Statement Variable Parameters .....	D-106
Table D-9	INC and EXC Statement Variable Parameters .....	D-106

---

Table D-10	SET Statement Function Parameters .....	D-108
Table D-11	SET Statement Diagnostic Parameters .....	D-112
Table D-12	INC and EXC Statement Diagnostic Parameters .....	D-112
Table D-13	SET Statement Event Parameters .....	D-113
Table D-14	INC and EXC Statement Event Parameters .....	D-113
Table D-15	SET Statement VTOC Scan Facility Parameters .....	D-117
Table D-16	INC and EXC Statement SMVSCFxx Parameter .....	D-117
Table E-1	Filter and Rule List Parameter Quick-Reference .....	E-1
Table F-1	Frequently Asked Questions .....	F-1

---

# About This Book

This book contains information that is common to all MAINVIEW® Storage Resource Manager products by BMC Software and is intended for storage administrators.

To use this book, you should be familiar with

- OS/390 operating system
- job control language (JCL)
- Interactive System Productivity Facility (ISPF)
- MAINVIEW navigation and infrastructure

Throughout this book, references to OS/390 support also include support for MVS and z/OS.

## How This Book Is Organized

This book is organized as follows. In addition, an index and glossary appear at the end of the book.

Chapter/Appendix	Description
Chapter 1, "Introducing MAINVIEW SRM"	helps you learn how to use MAINVIEW to monitor system performance
Chapter 2, "Using MAINVIEW SRM Operator Services"	describes the functions of SVOS, the started tasks that run all MAINVIEW SRM products
Chapter 3, "Defining the MAINVIEW SRM System"	provides an overview of global operational characteristics and system-wide functions of MAINVIEW SRM
Chapter 4, "Defining the SMFUNCxx Member and Functions"	describes how to use the SMFUNCxx parmlib member to define and activate functions; also describes in detail the USERVARS function, which is used by multiple components in MAINVIEW SRM

Chapter/Appendix	Description
Chapter 5, "Using Filter and Rule List Parameters"	describes how to use the FLSTxx and RLSTxx parmlib members to define filter and rule list parameters
Chapter 6, "Defining a Pool"	describes how to use the SMPOOLxx parmlib member to define pools
Chapter 7, "Using Variables"	describes how to use the SMVARSxx parmlib member to define variables SMVARSxx variables allow a user-specified name to be assigned any number of selection parameters.
Chapter 8, "Defining a Calendar"	describes how to use the SMCALSxx parmlib member to define non-working days in the MAINVIEW SRM system
Chapter 9, "Using Diagnostics"	describes how to use the SMDIAGxx parmlib member and provides an explanation and example of the tracing facility
Appendix A, "MAINVIEW SRM General Syntax Rules"	provides an overview of the allowable structures of the statements and parameters used in MAINVIEW SRM
Appendix B, "Processing SMF Records"	describes how to use the logging facilities to write activity messages to the SMF data set
Appendix C, "Altering Messages with the Message Exit"	provides instructions for modifying messages using the message exit
Appendix D, "Global Parameters"	provides a list of global system-parameter definitions that are used to define MAINVIEW SRM products
Appendix E, "Filter and Rule List Parameters"	provides a list of INC and EXC and SET statements that are used in the FLSTxx and RLSTxx members
Appendix F, "Troubleshooting"	provides a list of frequently asked questions developed by BMC Customer Support for MAINVIEW SRM

## Related Documentation

BMC Software products are supported by several types of documentation:

- online and printed books
- online Help
- release notes and other notices

In addition to this book and the online Help, you can find useful information in the publications listed in the following table. As "Online and Printed Books" on page xv explains, these publications are available on request from BMC Software.

Category	Document	Description
general	<i>MAINVIEW Products General Information</i>	provides an overview of the MAINVIEW environment and the products that it supports
MAINVIEW common documents	<i>OS/390 and z/OS Installer Guide</i> <i>MAINVIEW Installation Requirements Guide</i> <i>MAINVIEW Common Customization Guide Using MAINVIEW</i> <i>MAINVIEW Administration Guide</i> <i>Implementing Security for MAINVIEW</i>	provides instructions for installing, configuring, using, and administering MAINVIEW
MAINVIEW SRM customization documents	<i>MAINVIEW SRM Customization Guide</i>	provides instructions for configuring and customizing MAINVIEW SRM for OS/390
reference documents	<i>MAINVIEW SRM Reference Summary</i>	provides a listing and explanation of global system parameters, FLST/RLST parameters, and functions for all MAINVIEW SRM products
messages	<i>MAINVIEW SRM Messages</i>	provides hardcopy of messages that are also available online
product documents	<i>MAINVIEW SRM Allocation EasyPOOL User Guide and Reference</i> <i>MAINVIEW SRM Allocation EasySMS User Guide and Reference</i> <i>MAINVIEW SRM Allocation StopX37/II User Guide and Reference</i> <i>MAINVIEW SRM Automation User Guide</i> <i>MAINVIEW SRM Reporting User Guide</i> <i>MAINVIEW SRM Reporting Reference Manual</i>	provide product-specific information for MAINVIEW SRM products
supplemental documents	release notes, flashes, technical bulletins	provides additional information about MAINVIEW SRM products

## Online and Printed Books

The books that accompany BMC Software products are available in online format and printed format. If you are a Windows or Unix user, you can view online books with Acrobat Reader from Adobe Systems. The reader is provided at no cost, as explained in “To Access Online Books.” You can also obtain additional printed books from BMC Software, as explained in “To Request Additional Printed Books.”

---

## To Access Online Books

Online books are formatted as Portable Document Format (PDF) files. You can view them, print them, or copy them to your computer by using Acrobat Reader 3.0 or later. You can access online books from the documentation compact disc (CD) that accompanies your product or from the World Wide Web.

In some cases, installation of Acrobat Reader and downloading the online books is an optional part of the product-installation process. For information about downloading the free reader from the Web, go to the Adobe Systems site at <http://www.adobe.com>.

To view any online book that BMC Software offers, visit the support page of the BMC Software Web site at <http://www.bmc.com/support.html>. Select a product to access the related documentation.

## To Request Additional Printed Books

BMC Software provides printed books with your product order. To request additional books, go to <http://www.bmc.com/support.html>.

## Release Notes and Other Notices

Printed release notes accompany each BMC Software product. Release notes provide current information such as

- updates to the installation instructions
- last-minute product information

In addition, BMC Software sometimes provides updated product information between releases (in the form of a flash or a technical bulletin, for example). The latest versions of the release notes and other notices are available on the Web at <http://www.bmc.com/support.html>.

## Conventions

This section provides examples of the conventions used in this book and explains how to read ISPF panel-flow diagrams and syntax statements.



---

## General Conventions

This book uses the following general conventions:

Item	Example
information that you are instructed to type	Type <b>SEARCH DB</b> in the designated field.
specific (standard) keyboard key names	Press <b>Enter</b> .
field names, text on a panel	Type <b>the appropriate entry</b> in the <b>Command</b> field.
directories, file names, Web addresses	The BMC Software home page is at <b>www.bmc.com</b> .
nonspecific key names, option names	Use the HELP function key.  KEEPDICTIONARY option
calls, commands, control statements, keywords, parameters, reserved words	Use the SEARCH command to find a particular object.  The product generates the SQL TABLE statement next.
code examples, syntax statements, system messages, screen text	//STEPLIB DD  The table <i>table_name</i> is not available.
emphasized words, new terms, variables	The instructions that you give to the software are called <i>commands</i> .  In this message, the variable <i>file_name</i> represents the file that caused the error.
single-step procedures	»» To enable incremental backups, type <b>y</b> and press <b>Enter</b> at the next prompt.
GUI menu sequence	Choose <b>File =&gt; Open</b> .

This book uses the following types of special text:

**Note:** Notes contain important information that you should consider.

**Warning!** Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.

**Tip:** Tips contain useful information that may improve product performance or that may make procedures easier to follow.

---

## Syntax Statements

Syntax statements appear in Courier. The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2|KEYWORD3] KEYWORD4={YES|NO}  
file_name...
```

The following table explains conventions for syntax statements and provides examples:

Item	Example
Items in italic type represent variables that you must replace with a name or value. Use an underscore for variables with more than one word.	<i>dtbackup control_directory</i>
Brackets indicate a group of options. You can choose at least one of the items in the group, but none of them is required. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.	[ <i>table_name, column_name, field</i> ]
Braces enclose a list of required items. You must enter at least one of the items. Do not type the braces when you enter the item.	{ <i>DBD_name   table_name</i> }
A vertical bar means that you can choose only one of the listed items. In the example, you would choose either <i>commit</i> or <i>cancel</i> .	{ <i>commit   cancel</i> }
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.	<i>column_name . . .</i>

---

# Chapter 1     Introducing MAINVIEW SRM

This chapter provides a high-level introduction to the BMC Software MAINVIEW Storage Resource Manager (SRM) suite of products. The following topics are discussed:

Overview .....	1-2
MAINVIEW SRM Components. ....	1-2
Accessing MAINVIEW SRM .....	1-3
Using Easy Menus .....	1-4
Using EZCmd Menus .....	1-5

## Overview

The MAINVIEW SRM suite of storage products provides a wide range of services and functions to help you manage your entire enterprise storage system. Built on MAINVIEW architecture, MAINVIEW SRM employs a combination of the MAINVIEW standard ISPF panel interface, MAINVIEW windows interface, and MAINVIEW Explorer interface to provide easy, intuitive access to all of the storage data that you need.

To use MAINVIEW SRM to its fullest advantage, you should understand the key concepts and terms related to MAINVIEW. Refer to *Using MAINVIEW* for detailed information about the MAINVIEW architecture and interface.

## MAINVIEW SRM Components

The MAINVIEW SRM product suite is comprised of components shown in Table 1-1.

**Table 1-1      New Product Names**

New Product Names	Replaces Old Product Names
MAINVIEW SRM Allocation	EasyPOOL EasySMS StopX37/II
MAINVIEW SRM Automation	Enterprise Storage Automation
MAINVIEW SRM Reporting	EasyHSM StorageGUARD SG-Control

The MAINVIEW SRM Reporting product is comprised of the following collectors:

- application collector
- HSM collector
- performance collector
- space collector
- space alternate collector

## Accessing MAINVIEW SRM

When you start MAINVIEW using a 3270 display, the MAINVIEW Selection Menu is displayed. This menu is not supported in MAINVIEW Explorer PLEX mode.

To access MAINVIEW SRM, perform the following steps:

- Step 1** Select option **S** for **Storage Management** from the MAINVIEW Selection Menu.

The Storage Management Solutions submenu is displayed, as shown in Figure 1-1.

**Figure 1-1 Storage Management Solutions Submenu**

```

----- Storage Management Solutions -----
OPTION  ===>                                DATE  -- 2002/10/02
                                           TIME  -- 09:33:37
Management                                USERID -- JBRW
  1  MVS RM                               MAINVIEW Storage Resource Manager  MODE  -- ISPF 4.8

Operations
  A  AUTOMATION  MAINVIEW AutoOPERATOR
  E  ALERTS      Alert Management

General Services
  M  MESSAGES    Messages and Codes
  P  PARMS       Parameters and Options

```

- Step 2** Select option **1** for **MAINVIEW Storage Resource Manager** from the Storage Management Solutions submenu.

The EZSRM Menu is displayed, as shown in Figure 1-2 on page 1-4.

**Figure 1-2 EZSRM Menu**

```

02OCT2002 09:36:19 ----- MAINVIEW WINDOW INTERFACE(V4.1.03)-----
COMMAND ==> SCROLL ==> PAGE
CURR WIN ==> 1 ALT WIN ==>
>W1 =EZSRM=====SJSGCAS4=*=====02OCT2002==09:36:19===MVS RM===D===1
EZSRM Menu

DASD Analysis      +-----+      Tape & HSM Views
                   |         |      > Tape
                   | Place cursor on |      > DFSMS/HSM
                   | menu item and   |
                   | press ENTER    |
+-----+
> Configuration
> Space Utilization
> Performance
> Data Set Management
> Group & Pool Analysis
> Volume Analysis

Administration
> Application Management
> Parmlib Members
. Functions
. Component Status

Tools and Menus
> Automation
> Audit Log
. StopX37/II Activity
> SMF Report Library
. Batch Report Facility
. MAINVIEW Messages
. View List
. Return....

```

## Using Easy Menus

An Easy Menu consists of a series of options, all of which hyperlink to data views or other menus that are specific to the current option. The names of Easy Menus are prefixed with the letters EZ.

Options that are preceded by a period (.) display system data. Options that are preceded by a greater-than sign (>) display either pop-up menus in the center of the window or other menu views from which you can access information.

The EZSRM Menu (Figure 1-2) is the initial display when you access MAINVIEW SRM. All options on this menu hyperlink to high-level views or submenus. Easy Menu options have descriptive, intuitive names that correspond to an aspect of the storage system, enabling you to use MAINVIEW SRM quickly and easily without having to learn the names and functions of specific views.

## Using EZCmd Menus

EZCmd menus provide you with options for viewing information about a selected pool, volume, group, data set, and so on. You can access EZCmd menus by selecting the first data field in a row. The resulting EZCmd menu contains links for all views or actions related to the type of resource that you selected. For example, the Pool EZCmd menu (EZPOOL) is displayed when you select the first field from any MAINVIEW SRM view that is displaying pool information.





---

## Chapter 2      Using MAINVIEW SRM Operator Services

This chapter provides an overview of the operator services used in MAINVIEW SRM. The following topics are discussed:

Overview .....	2-2
Activating System Software in OS/390 .....	2-2
Starting and Stopping MAINVIEW SRM .....	2-3
Preparing to Start SVOS .....	2-4
Parameter Descriptions .....	2-5
Starting SVOS .....	2-8
Starting MAINVIEW SRM Components and Subcomponents .....	2-8
Stopping MAINVIEW SRM Components .....	2-12
Stopping SVOS .....	2-13
Starting and Stopping MAINVIEW SRM with Other Products .....	2-13
Communicating with MAINVIEW SRM Components .....	2-14
Common Command Structure .....	2-14
Controlling MAINVIEW SRM Components .....	2-14
Communicating with MAINVIEW SRM Components .....	2-14
SVOS Loader Console Commands .....	2-16
SVOS Component Console Commands .....	2-21
Tracking MAINVIEW SRM Activity .....	2-25
Audit Log Views .....	2-26
Audit Log Records .....	2-26

## Overview

MAINVIEW SRM operator services (SVOS) is a started task that runs all MAINVIEW SRM products. SVOS is the product address space (PAS) that manages the connection and communication with the common address space (CAS). The SVOS PAS provides

- connection for MAINVIEW SRM views
- dynamic activation and deactivation of MAINVIEW SRM components
- communication with MAINVIEW SRM components

In addition to explaining these functions in detail, this chapter includes information about

- SVOS loader console commands
- SVOS component console commands

## Activating System Software in OS/390

There are two methods of activating system software products:

### **Static Activation**

Usually, this method uses SMP to link a USERMOD into the operating system. The modification is actually applied to system libraries and an IPL usually is required to activate the product. While this is a safe method of activating system software products, there are serious system maintenance drawbacks. Manual intervention is required by system programmers whenever maintenance is applied to modified CSECTS. Just scheduling the SMP work, and an IPL, often can be a problem.

### **Dynamic Activation**

This method requires a sophisticated activation process. SVOS is a started task that performs this process for MAINVIEW SRM components. SVOS loads and controls the various component program modules in the OS/390 operating system. This started task eliminates the system maintenance problems that are involved with static modification.

# Starting and Stopping MAINVIEW SRM

All MAINVIEW SRM components are controlled and managed with the MAINVIEW SRM common component SVOS address space. All MAINVIEW SRM components are started within this common address space. The SVOS address space also controls the connection to the MVI CAS and, in MAINVIEW terminology, is the PAS for MAINVIEW SRM.

When the MAINVIEW SRM common PAS, SVOS, is started, the default is to start all licensed components at startup.

The global system parameter BBI3\_SSID specifies the CAS subsystem name to which the SVOS PAS should connect. Because the MAINVIEW Infrastructure (MVI) connection occurs during SVOS startup, SVOS will not start if BBI3\_SSID is not specified. To update the value of BBI3\_SSID, SVOS must be stopped and restarted; it cannot be refreshed. The CAS subsystem name is specified in the SSID parameter on the PARM keyword for the CAS JCL EXEC statement.

**Tip:** You can run the StopX37/II product without the MAINVIEW interface, and in this case SVOS does not require the BBI3\_SSID keyword to start. For more information about the keyword, see the STOPX37II= keyword of the SMMSYS00 parmlib member.

## Preparing to Start SVOS

Sample JCL for SVOS is in the SVOS member of the *?prefix.BBSAMP* data set. Table 2-1 describes the JCL DD statements and start-up options passed in the JCL PARM= statement.

**Note:** The START command and the EXEC label in the procedure being started *must* agree for the system to start.

**Table 2-1 Execution JCL DD Statements and Start-Up Options (Part 1 of 2)**

Statement	Use
ACTLOG01 - ACTLOGxx	specifies the audit log files that MAINVIEW SRM components use to track activity of various functions The files are defined during MAINVIEW SRM customization and cannot be shared by SVOS address spaces on other systems.
BBACTDEF	specifies the data sets that contain definitions required by the MAINVIEW Infrastructure These data sets are created during the SM install. If you run StopX37/II without MAINVIEW, you do not need this DD statement, and if you include it, MAINVIEW SRM ignores it.
BBIPARM	specifies the data sets that contain customizable parameters for the MAINVIEW Infrastructure These data sets are created during the SMP installation. If you run StopX37/II without MAINVIEW, you do not need this DD statement, and if you include it, MAINVIEW SRM ignores it.
BBSECURE	specifies the data sets that contain definitions required by the Security feature of the MAINVIEW Infrastructure These data sets are created during the SMP installation. See the manual entitled <i>Implementing Security for MAINVIEW Products</i> for more information.
BBSLIB	specifies the data sets that contain skeleton JCL members used by the MAINVIEW SRM Automation component These data sets are created during the SMP installation. If you run StopX37/II without MAINVIEW, you do not need this DD statement, and if you include it, MAINVIEW SRM ignores it.
BBVDEF	specifies the data sets that contain view definitions used by the MAINVIEW Infrastructure These data sets are created during the SMP installation. If you run StopX37/II without MAINVIEW, you do not need this DD statement, and if you include it, MAINVIEW SRM ignores it.
EXEC	specifies the program name for SVOS (PGM=ETILOADR)
PROC	procedure statement for the started task
SNAP	specifies a output sequential data set to be used by SVOS when generating diagnostic dumps due to abend or other error situation Diagnostic data produced to this output DD might be required by BMC Software Customer Support to resolve problems. Make sure that this output is to a permanent medium.

**Table 2-1 Execution JCL DD Statements and Start-Up Options (Part 2 of 2)**

Statement	Use
STEPLIB DD	<p>defines one or more input partitioned data sets</p> <p>These DD statements allocate various load libraries that are required for executing SVOS and BMC Software products. All libraries that are allocated to STEPLIB must be authorized. SVOS is divided into four load modules: LODR410\$ (ETILOADR), LODR411\$, LODR412\$, and DISR410\$ (ETIDYNAM). The BMC Software products are also in this library.</p>
SVOSPARM	<p>specifies the data set containing customizable parameters for the MAINVIEW SRM system</p> <p>These data sets are created during the SMP installation, and were modified during MAINVIEW SRM customization.</p>
SYSABEND	<p>defines a sequential output data set that is used for diagnostic dump information</p> <p>This data might be required by BMC Software Customer Support to resolve problems. You can also produce diagnostic data to the SNAP DD.</p>
SYSLIB DD	<p>specifies a cataloged data set name for the LPALIB library concatenations that are to be allocated at SVOS startup as a default</p> <p>LPALIB data sets must be the same as they were when the last system IPL was performed with a CLPA or an MPLA. A maximum of three data sets can be concatenated. A SYSLIB DD statement in the JCL overrides a SYSLIB global parameter.</p> <p><b>Notes:</b> You must specify either the SYSLIB global parameter or include the LPALIB in the execution JCL.</p> <p>You can specify a cataloged data set name for the LPALIB library concatenations that are to be allocated at SVOS startup as a default in the SYSLIB, SYSLIB2, and SYSLIB3 system parameters. You can override these parameters with an SYSLIB DD statement in the JCL. LPALIB data sets must be the same as they were when the last system IPL was performed with a CLPA or an MPLA. A maximum of three data sets can be concatenated.</p>
SYSOUT	<p>defines a sequential output data set that is used for listing output from system utilities such as SORT</p> <p>You can use a nonexistent SYSOUT class to cause the SYSOUT file to be purged after you terminate SVOS.</p>
SYSPRINT DD	<p>defines a sequential output data set that is used for listing actions that were taken during dynamic activation processing</p> <p>You can use a nonexistent SYSOUT class to purge the SYSPRINT file after you terminate SVOS. The DCB information is DCB=(LRECL=80,RECFM=FB).</p>
SYSTEM	<p>defines a sequential output data set that is used for listing output from system utilities such as SORT</p> <p>You can use a nonexistent SYSTEM class to cause the SYSTEM file to be purged after you terminate SVOS.</p>

## Parameter Descriptions

During initialization, SVOS processes the parameter information that is passed with the PARM field on the EXEC statement. These parameters are used to pass installation-dependent options to the program. A list of the parameters and their defaults follows. All parameters are separated from their values with an equal sign (=).

## FORCE Parameter

**Function:** The FORCE parameter determines whether the SVOS FORCE command can be issued.

**Warning!** If used improperly, the FORCE command can corrupt system control blocks and load modules.

If FORCE=NO is specified, the FORCE command is deactivated and cannot be used.

**Default:** FORCE=YES

**Format:** YES or NO, Y or N

## NOVERIFY Parameter

**Function:** The NOVERIFY parameter determines whether the SVOS START and STOP commands can use the NOVERIFY operand. The activation or deactivation procedure normally validates the pointers that are being changed in system control blocks and load modules. If NOVERIFY=YES is specified and the NOVER operand is requested on START/STOP, SVOS modifies storage without validating the current pointers.

**Default:** NOVERIFY=YES

**Format:** YES or NO, Y or N

## SUBCHAR Parameter

**Function:** The SUBCHAR parameter can be used in addition to MODIFY commands to enter SVOS commands from the console. The slash (/) is the default character, but this operand can be used to select any character as the subsystem character. If SUBCHAR=NONE is specified, only the MODIFY form of the loader commands are processed.

**Default:** SUBCHAR=

**Format:** Any valid subsystem command character or NONE

**Note:** When you are using the multiple subsystem feature of SVOS, the subsystem character should be unique for each SVOS task that is started.

**SUBSYS Parameter**

**Function:** The SUBSYS parameter defines the subsystem name that identifies SVOS to the system. If the default subsystem name needs to be overridden, specify the new four-character subsystem name by using the SUBSYS parameter.

**Default:** SUBSYS=ETIS

**Format:** Any valid four-character subsystem name.

**CANCEL Parameter**

**Function:** The CANCEL parameter allows SVOS to run as CANCELABLE or NON-CANCELABLE. The system may be exposed to error conditions if SVOS is eligible to be cancelled. OS/390 will not honor a cancel request by the operator if SVOS is running as NON-CANCELABLE. This parameter allows SVOS to be shut down without error and resolve all changes made to the system. If YES is coded during SVOS initialization, a warning message is sent to the operator to respond either YES (allow) or NO (prevent) further initialization. If NO is specified, no message is sent and SVOS continues initialization as normal and SVOS runs as NON-CANCELABLE.

**Default:** CANCEL=NO

**Format:** YES or NO, Y or N

**SYS Parameter**

**Function:** Specifies the suffix of the SMMSYSxx member to use when starting SVOS. The SMMSYSxx member must exist in the MAINVIEW SRM system parameter data set specified in the SVOSPARM DD statement.

**Default:** SYS=00

**Format:** two-position suffix

## Starting SVOS

SVOS is started with the following OS/390 command:

```
S SVOS
```

After SVOS has successfully started, the following message is displayed on the console:

```
SVO0106 RELEASE v.r.m OF SVOS HAS BEEN STARTED
```

At this point the SVOS address space is up and can begin to activate components. This part of the MAINVIEW SRM product, the framework under which all MAINVIEW SRM components are started and controlled, is referred to as the loader. After successfully starting, the loader automatically starts the SVOS component. The SVOS component, like the loader, is common to all MAINVIEW SRM components. The SVOS component controls the reading of the MAINVIEW SRM system parameters from the parmlib data set, as discussed in Chapter 3, “Defining the MAINVIEW SRM System,” and prepares the environment for all other components to be started.

After the SVOS component is active, the following message is displayed on the console:

```
SVO0610I MAINVIEW SRM SVOS 7.2.00 HAS BEEN STARTED ON  
EM72
```

SVOS is designed to start successfully, even if parameter errors are present. If there are parameter errors, access the Parmlib Members option (ADMEN view), correct the errors in the appropriate members, and perform a refresh. You can start the other components of MAINVIEW SRM after you refresh the corrected members.

If there is attempt to start another component after SVOS has had a parameter error, the startup attempt will fail and messages will refer to the SVOS parameter error.

## Starting MAINVIEW SRM Components and Subcomponents

By default, all licensed components are started when SVOS is started. Optionally, you can specify one or more components or subcomponents to *not* start by setting one or more of the keywords in Table 2-2 to N.



**Table 2-2 Startup Override Keywords**

<b>System Keyword</b>	<b>Description</b>
START_ALL=Y/N	specifies whether any components other than SVOS should be started
START_ALLOC=Y/N	specifies whether the Allocation component should be started
START_AUTO=Y/N	specifies whether the Automation component should be started
START_EHSM=Y/N	specifies whether the HSM collector should be started
START_EPOOL=Y/N	specifies whether the EasyPOOL subcomponent should be started
START_ESMS=Y/N	specifies whether the EasySMS subcomponent should be started
START_RPRT=Y/N	specifies whether the Reporting component should be started
START_SGA=Y/N	specifies whether the SG-Auto subcomponent should be started
START_SGC=Y/N	specifies whether the application collector should be started
START_SGD=Y/N	specifies whether the space collector should be started
START_SGP=Y/N	specifies whether the performance collector should be started
START_X37=Y/N	specifies whether the StopX37/II subcomponent should be started

After you have prevented a component or subcomponent from starting, you can manually start it by using

- the Component Status option on the EZSRM menu
- the start commands listed in Table 2-3

**Table 2-3 Component Start Commands**

<b>Component</b>	<b>Subcomponent</b>	<b>Command</b>
SVOS	n/a	/S SVOS,SUF=00
SG-Auto	n/a	/S SVSGA
Allocation	EasySMS EasyPOOL StopX37/II	/S SVALLOC
Automation	n/a	/S SVESA
Reporting	application collector HSM collector performance collector space collector space alternate collector	/S SVSGC /S SVHSM /S SVSGP /S SVSGD /S SVSGDn <i>(n is a number, in the range 1–8, that identifies the alternate data collector)</i>

The STATUS command is automatically issued after all necessary subcomponents have been started.

After a successful start up of the MAINVIEW SRM component, the following messages are displayed on the console indicating that MAINVIEW SRM is active:

```
SVO0106 RELEASE v.r.m OF SVOS HAS BEEN STARTED
SVO0610 MAINVIEW SRM/SGD REL v.r.m HAS BEEN STARTED ON sybsys
SVO0610 MAINVIEW SRM/HSM REL v.r.m HAS BEEN STARTED ON sybsys
SVO0610 MAINVIEW SRM/SGP REL v.r.m HAS BEEN STARTED ON sybsys
SVO0610 MAINVIEW SRM/SGA REL v.r.m HAS BEEN STARTED ON sybsys
SVO0610 MAINVIEW SRM/SGC REL v.r.m HAS BEEN STARTED ON sybsys
SVO0610 MAINVIEW SRM/SVESA REL v.r.m HAS BEEN STARTED ON sybsys
SVO0610 MAINVIEW SRM/ALLOC REL v.r.m HAS BEEN STARTED ON sybsys
```

In these messages, *v.r.m* is the version, release, and modification number and *sybsys* is the sub-system name that is used in SUBSYS parameter in SVOSJCL (the default is ETIS).

## Viewing the Status of Components and Subcomponents

You can use the Component Status view (COMPSTAT) or the Component Summary view (COMPSUMM) to see the status of components and subcomponents. You can access these views by choosing **Component Status** from the EZSRM menu or by typing the view name on the command line.

COMPSTAT displays a list of all *subcomponents* on your system, along with the active status and license information for each subcomponent. Figure 2-1 on page 2-11 shows the COMPSTAT view.

To filter the COMPSTAT view to display only active components, invoke COMPSTAT \* Y. To filter the COMPSTAT view to display only licensed components, invoke COMPSTATLIC.

COMPSUMM displays a list of all *components* on your system, along with the active status and license information for each component. Figure 2-2 on page 2-11 shows the COMPSUMM view. To view a list of subcomponents that comprise the component and the status of each, click on the component name.

**Tip:** You can access COMPSUMM from the COMPSTAT view by placing the cursor in the Component field and pressing enter.

You can activate or deactivate all components except SVOS from COMPSTAT and COMPSUMM as follows:

- Type the A or S line commands to activate/start components.
- Type the I or P line commands to inactivate/stop components

**Figure 2-1 Component Status (COMPSTAT)**

```

24OCT2002 10:50:23 ----- MAINVIEW WINDOW INTERFACE(V4.1.04)-----
COMMAND ==>                                SCROLL ==> PAGE
CURR WIN ==> 1          ALT WIN ==>
>W1 =COMPSTAT=====SJSGCAS4=*=====24OCT2002==08:57:16====MVSRM====D====11
CMD Subcomponent      Component  Component  Lic Started  SSI      Server
--- -----
Application Collector  ACTIVE    Reporting  Y   SVSGC    SJSGCAS4  SVW72JB
EasyPOOL              ACTIVE    Allocation Y   SVALLOC  SJSGCAS4  SVW72JB
EasySMS               ACTIVE    Allocation Y   SVALLOC  SJSGCAS4  SVW72JB
HSM Collector         ACTIVE    Reporting  Y   SVHSM    SJSGCAS4  SVW72JB
Performance Collector ACTIVE    Reporting  Y   SVSGP    SJSGCAS4  SVW72JB
Space Collector       ACTIVE    Reporting  Y   SVSGD    SJSGCAS4  SVW72JB
StopX37/II           ACTIVE    Allocation Y   SVALLOC  SJSGCAS4  SVW72JB
StopX37/II Only      NOT ACTIVE Allocation Y   X37II    SJSGCAS4  SVW72JB
Storage Automation    ACTIVE    Automation Y   SVESA    SJSGCAS4  SVW72JB
SGAuto               NOT ACTIVE SGAuto     Y   SVSGA    SJSGCAS4  SVW72JB
SVOS Operator Services ACTIVE    Common     Y   SVOS     SJSGCAS4  SVW72JB

```

**Tip:** Because the SVOS component must be licensed and started to be connected to MAINVIEW, it always appears as LICENSED and ACTIVE. The LICENSED COMPONENT field contains the value COMMON, which indicates that this product is licensed with any other component.

**Figure 2-2 Component Summary (COMPSUMM)**

```

24OCT2002 11:01:19 ----- MAINVIEW WINDOW INTERFACE(V4.1.04)-----
COMMAND ==>                                SCROLL ==> PAGE
CURR WIN ==> 1          ALT WIN ==>
W1 =COMPSUMM=====SJSGCAS4=*=====24OCT2002==11:01:19====MVSRM====D====5
CMD Component  Component  # Sub    Num  Num Lic SSI      Server  Sysplex Act
--- Name       Status    Comps Active Inact Ind System  Name    Name    Ind
Allocation NOT ACTIVE    4          4 Y   SJSGCAS4  SVW72JBS BBPLEX01 N
Automation NOT ACTIVE    1          1 Y   SJSGCAS4  SVW72JBS BBPLEX01 N
Common       ACTIVE      1          1 Y   SJSGCAS4  SVW72JBS BBPLEX01 Y
Reporting    ACTIVE      4          3 Y   SJSGCAS4  SVW72JBS BBPLEX01 *
SGAuto       NOT ACTIVE    1          1 Y   SJSGCAS4  SVW72JBS BBPLEX01 N

```

## Stopping MAINVIEW SRM Components

Stopping SVOS stops all other components in the selected SVOS server. The SVOS component itself then stops, which disconnects it from MAINVIEW. After this occurs, the MAINVIEW SRM product is no longer accessible and appears as INACTIVE on the PLEX view of the PlexManager.

You can stop individual components using the Component Status option on the EZSRM menu or you can type the appropriate command on the console to stop the components. Table 2-4 lists the shutdown commands you can use.

Stopping the SVOS component or the SVOS address space causes all subcomponents to stop prior to the SVOS stopping.

**Table 2-4 Shutdown Commands**

Component	Subcomponent	Command
Reporting	space collector performance collector application collector HSM collector	/P SVSGD /P SVSGP /P SVSGC P SVHSM
Allocation	EasySMS EasyPOOL StopX37/II	/P SVALLOC
SG-Auto	SG-Auto	/P SVSGA
Automation	n/a	/P SVESA
SVOS	n/a	/P SVOS

In these commands, / is the SVOS command character that is specified in SUBCHAR parameter in SVOS JCL.

After a successful shutdown of the MAINVIEW SRM components, the following message are displayed on the console:

```
SVO0620 MAINVIEW SRM/SGD REL v.r.m HAS BEEN STOPPED ON subsys
SVO0620 MAINVIEW SRM/SGP REL v.r.m HAS BEEN STOPPED ON subsys
SVO0620 MAINVIEW SRM/SGA REL v.r.m HAS BEEN STOPPED ON subsys
SVO0610 MAINVIEW SRM/SGC REL v.r.m HAS BEEN STOPPED ON subsys
SVO0610 MAINVIEW SRM/SVESA REL v.r.m HAS BEEN STOPPED ON subsys
SVO0620 MAINVIEW SRM/ALLOC REL v.r.m HAS BEEN STOPPED ON subsys
```

In these messages, *v.r.m* is the version, release, and modification number and *subsys* is the sub-system name that is used in SUBSYS= parameter in SVOSJCL (the default is ETIS).

## Stopping SVOS

To stop SVOS and all active components, type

```
/ SHUTDOWN
```

In this command, / is the SVOS command character that is specified in SUBCHAR parameter in SVOS JCL.

The SHUTDOWN command stops SVOS and any BMC Software products that are running under SVOS. No SVOS messages are written to the console on a successful shutdown.

## Starting and Stopping MAINVIEW SRM with Other Products

MAINVIEW SRM dynamically hooks into the operating system when it starts; when shut down, MAINVIEW SRM unhooks itself in an orderly manner. If other system management products are started that hook into the operating system in a similar manner, problems can occur on shutdown of those products or MAINVIEW SRM, depending on the order of startup and shutdown.

Contact BMC Software Customer Support for the most current information on compatibility with other system management products. In general, MAINVIEW SRM should be the last product started and the first stopped. In all cases, products should be stopped in the reverse order of startup; that is, if product A is started first, followed by MAINVIEW SRM, MAINVIEW SRM should be stopped first, followed by product A.

**Note:** PSP releases 115 and earlier must be started after MAINVIEW SRM. Softworks has zaps to fix this problem; the zap IDs are SWAG011.200 and SWAG012.200. (These zaps have been added to PSP 116.) After these zaps are applied, or if you are using PSP 116 or earlier, MAINVIEW SRM can be started after PSP is initialized.

# Communicating with MAINVIEW SRM Components

SVOS manages common storage, controls any required subtasks, and performs console command processing. This function of SVOS offers

- a common command structure
- control of MAINVIEW SRM components

## Common Command Structure

SVOS uses a common command structure that allows all MAINVIEW SRM components to respond to one set of commands. The base command is the same regardless of the component; only the component identifier code is specific to a component.

## Controlling MAINVIEW SRM Components

Using SVOS commands, MAINVIEW SRM components can be activated or deactivated easily. Also, dynamically changing component options is easily accomplished using SVOS.

Table 2-5 identifies each component or set of components for MAINVIEW SRM and the component or subcomponent identifier for each.

**Table 2-5**                      **MAINVIEW SRM Components and Subcomponents**

Component	Subcomponent	Identifier
SVOS	n/a	SVOS
MAINVIEW SRM Reporting	HSM collector space collector performance collector application collector	SVHSM SVSGD SVSGP SVSGC
MAINVIEW SRM Allocation	EasyPOOL EasySMS StopX37/II	SVALLOC
MAINVIEW SRM Automation	n/a	SVESA
SG-Auto	n/a	SVSGA

## SVOS Command Syntax Conventions

The following syntax conventions apply to SVOS commands. They are used throughout this chapter to explain SVOS commands and the optional parameters.

- Square brackets [ ] indicate that the parameters within the brackets are optional. Type only information that is shown within the brackets. Do not type the brackets.
- Braces { } indicate that you must choose among items found within the braces. Use only one item. Do not type the braces.
- A vertical bar | indicates that you must choose among the items separated by the bar. The bar is used in conjunction with the brackets and braces. Use only one item. Do not type the bar.
- Uppercase indicates that the field must be typed as shown. You can type it in either uppercase or lowercase.
- Lowercase indicates that the operand must be supplied to the command. You can type it in either uppercase or lowercase.
- Commands and operands can be separated with either blanks or commas.

### Using MODIFY Commands

After SVOS has been started, MODIFY commands can be used to enter commands from the console. The following example shows how to use the MODIFY command:

```
F SVOS ,STATUS SVSGD
F SVOS ,STOP SVALLOC
```

### Using a Subsystem Character

The SVOS subsystem character can be used to enter commands from the console. The slash (/) is the default character, but the SUBCHAR parameter can be used to override the default subsystem character. To disable the SVOS subsystem character and allow the use of the MODIFY command only, specify SUBCHAR=NONE in the parameter field in the SVOS started task.

By using the subsystem character, you do not need to remember the started task name that is used by the MODIFY commands. To get the same results as shown in the MODIFY command example, type the following commands on the console:

```
/STATUS SVALLOC  
/START SVSGP  
/STOP SVSGD
```

## SVOS Loader Console Commands

Table 2-6 lists the SVOS loader console commands. This section describe each command.

**Table 2-6**            **SVOS Loader Commands**

Command	Synonym	Purpose
DUMP	none	terminates SVOS with an abend
FORCE	none	forces SVOS to perform requested procedure
SHUTDOWN	none	terminates SVOS and all products that are running under it
START	S	initiates the requested product
STATUS	STAT	lists status information for one or all products
STOP	P	deactivates the requested product

### DUMP Command

**Purpose:** The DUMP command forces SVOS to terminate with a user 1000 abend.

**Syntax:** DUMP

**Comments:** Under normal conditions, this command should not be required. Use it only at the request of BMC Software Customer Support.

**Examples:** At the request of BMC Software Customer Support, you can enter either of the following console commands to force SVOS to abend:

```
/DUMP
```

*or*

```
F SVOS ,DUMP
```



## FORCE Command

**Purpose:** The FORCE command forces SVOS to go through activation or deactivation processing for a product. It also can be used to reinitialize the SVOS subsystem tables.

**Syntax:** FORCE { \*SSVT | \*SSCVT | *product\_name* [START | STOP] }

**Comments:** Under normal circumstances, this command should not be required. Use the \*SSVT and \*SSCVT operands only at the request of BMC Software Customer Support.

The *product\_name* operand should be used only when a START or STOP command cannot complete. This unusual condition should occur only when a product has been partially activated. When processing the *product\_name* operand, if neither START nor STOP has been requested, SVOS simply marks the product as inactive.

**Examples:** If SVOS had been cancelled during SVALLOC initialization, either of the following console commands will force SVOS to restart SVALLOC:

```
/FORCE SVALLOC,START
```

*or*

```
F SVOS,FORCE SVALLOC,START
```

At the request of BMC Software Customer Support, enter either of the following console commands to force SVOS to reinitialize the SSVT for the ETIS subsystem. SVOS uses this subsystem to keep track of the status of BMC Software products under its control.

```
/FORCE *SSVT
```

*or*

```
F SVOS,FORCE *SSVT
```

## SHUTDOWN Command

**Purpose:** The SHUTDOWN command forces SVOS to deactivate all BMC Software products that were activated by SVOS, and then SVOS terminates.

**Syntax:** SHUTDOWN

**Examples:** To deactivate all products that are running under SVOS and terminate SVOS, use one of the following console commands:

/ SHUTDOWN

*or*

F SVOS , SHUTDOWN

## START Command

**Purpose:** The START command informs SVOS to perform activation for the requested component.

**Syntax:** START *component\_name* [NOVER]

**Synonym:** S

**Comments:** When SVOS is started, all components and subcomponents are started unless a system keyword specifies otherwise (see Table 2-2, “Startup Override Keywords,” on page 2-9).

Use the NOVER operand only at the request of BMC Software Customer Support. The NOVER operand bypasses the verification procedure that is normally performed before storage is modified. If you specify START NOVER, then STOP processing defaults to NOVER also and all enqueue protection is bypassed on the REFRESH and STOP command for the product. To disable this operand, specify NOVERIFY=NO on the EXEC statement in the SVOS JCL.

## STATUS Command

**Purpose:** The STATUS command instructs SVOS to list information about MAINVIEW SRM components. The STATUS command is automatically issued after all of the necessary subcomponents have been started.

**Syntax:** STATUS [*component\_name* | \* | VMAP] [LIST | VER] [ALL]

**Synonym:** STAT

**Comments:** A STATUS command with no parameters lists the BMC Software products that are currently active.

Specify a *product\_name* to request information about only that product.

Specify VER to display the current product version rather than the selection module information.

Specify the LIST operand to display information that SVOS uses to control a product. Under normal conditions, LIST is only used by BMC Software Customer Support.

If ALL is specified, the status for all SVOS subsystems is displayed for each product, using the LIST or VER option as supplied.

**Note:** If VMAP is specified, *product\_name* cannot be specified. ALL can be specified to produce a VMAP of all SVOS subsystems. If you specify LIST or VER, they are ignored.

To display a virtual storage map of all SVOS subsystems, issue the following console command:

```
/STATUS VMAP ALL
```

To display a virtual storage map of all SVOS subsystems and to see the version number of SVOS, issue the following console command:

```
/STATUS VMAP ALL VER
```

To display the status of all products across all SVOS subsystems, issue the following console command:

```
/STATUS ALL
```

## STOP Command

**Purpose:** The STOP command instructs SVOS to deactivate the requested product.

**Syntax:** STOP *component\_name* [VER | NOVER]

**Synonym:** P

**Comments:** Use the SHUTDOWN command to deactivate all MAINVIEW SRM components and terminate SVOS.

Under normal condition, the VER and NOVER operands are not needed; use them only at the request of BMC Software Customer Support. Unless the associated START command requested NOVER, STOP commands perform a verification by default before changing memory. To disable this operand, specify NOVERIFY=NO on the EXEC statement in the SVOS JCL.

**Examples:** If you want to deactivate SVSGD, issue one of the following console commands:

```
/STOP SVSGD
```

*or*

```
F SVOS,STOP SVSGD
```

## SVOS Component Console Commands

In addition to controlling MAINVIEW SRM through the MAINVIEW interface, you can control MAINVIEW SRM by operator commands that you enter through the console. Commands are directed to SVOS, which relays the commands to the MAINVIEW SRM component for execution.

You can direct commands to SVOS in these ways:

- use the component identifier that is defined to SVOS with the SUBCHAR parameter (see “Using a Subsystem Character” on page 2-15). For example:

```
/SVOS R,SYS=00
```

In this command, / is the subsystem character.

- use the OS/390 MODIFY command. For example:

```
F SVOS,SVOS R,SYS=00
```

In this command, SVOS is the name of the MAINVIEW SRM operator services job.

The SVOS component console commands that are common to all MAINVIEW SRM components are:

- DISPLAY
- REFRESH
- SET
- SVOS CONFIG,REFRESH
- SVOS CONFIG,DIAG(DUMP|*volser*|*device address*)

Additional commands that are used to support a specific component of MAINVIEW SRM are documented in the user manual for the component.

### DISPLAY Commands

Use the DISPLAY command to show the current parameter values and members in the MAINVIEW SRM component that is running. Table 2-7 on page 2-22 describes the command syntax. The following abbreviations are valid for the DISPLAY console command:

- D
- DISP
- DISPL
- DISPLAY

**Table 2-7      DISPLAY Command Syntax**

<b>Command Syntax</b>	<b>Command Function</b>
D,CAL,ALL	displays non-working days for all years in the calendar member
D,CAL= <i>year</i>	displays non-working days for a specific year
D,DIAG,ALL	displays all user-defined diagnostic members
D,DIAG= <i>diagnostic name</i>	displays a single user-defined diagnostic member
D,EVNT,ALL	displays information about all event IDs
D,EVNT= <i>eventid</i>	displays information about a specific event ID
D,FUNC,ACT	displays information about all active functions
D,FUNC,ALL	displays information about all functions
D,FUNC= <i>function name</i>	displays information about a specific function
D,FUNC= <i>function name</i> ,FLST	displays information about a specific function's filter list
D,FUNC= <i>function name</i> ,RLST	displays information about a specific function's rule list
D,POOL,ALL	displays information about all pools
D,POOL= <i>pool name</i>	displays information about a specific pool
D,SPOL,ALL	displays information about all DFSMS-managed subpools
D,SPOL= <i>subpool</i>	displays information about a specific DFSMS-managed subpool
D,SYS	displays the current MAINVIEW SRM system definition For example: F SVOS,SVOS D,SYS
D,VARS,ALL	displays all user-defined variables
D,VARS= <i>variable name</i>	displays a single user-defined variable

**REFRESH Command**

Use the REFRESH command to rebuild MAINVIEW SRM parameters from the members in the parameter data set. Table 2-8, “REFRESH Command Syntax,” on page 2-23 describes the command syntax. The following abbreviations are valid for the REFRESH console command:

- R
- REF
- REFRL
- REFRESH

**Table 2-8 REFRESH Command Syntax**

<b>Command Syntax</b>	<b>Command Function</b>
R,CAL=xx	causes the SMCALSxx member to be read and all calendar parameters to be rebuilt
R,CRIT=xx	refreshes the SMCRTxx member
R,DIAG=xx	causes the SMDIAGxx member to be read and all diagnostic parameters to be rebuilt
R,EVNT=xx	causes the SMEVNTxx member to be read and all event parameters to be rebuilt
R,FUNC=xx	causes the SMFUNCxx member to be read and all function parameters to be rebuilt, including SMFLSTxx and SMRLSTxx
R,FUNC=function name,FLSTxx	causes the specified filter list (SMFLSTxx) for the named function to be rebuilt from the parameter library member
R,FUNC=function name,RLSTxx	causes the specified rule list (SMRLSTxx) for the named function to be rebuilt from the parameter library member
R,POOL=xx	causes the SMPPOOLxx member to be read and all pool parameters to be rebuilt
R,SPOL=xx	causes the SMSPOLxx member to be read and all SMS subpool parameters to be rebuilt
R,SYS=xx	causes the complete MAINVIEW SRM component to be rebuilt and all parameters refreshed, based on the SMMSYSxx member that is identified by specified suffix
R,VAR=xx	causes the SMVARsxx member to be read and all variables to be rebuilt

## SET Command

Use the SET command to *temporarily* alter the values of certain MAINVIEW SRM parameters.

**Warning!** Changes you make with the SET command are temporary. The parameter values are reset to those defined in the MAINVIEW SRM parmlib members after MAINVIEW SRM is stopped and restarted.

Table 2-9 on page 2-24 describes the command syntax. The following abbreviations are valid for the SET console command:

- T
- SET

**Table 2-9 SET Command Syntax**

<b>Command Syntax</b>	<b>Command Function</b>
<i>T, FUNC=function name, ACTIVE=YES/NO</i>	temporarily changes the status of the specified function
<i>T, FUNC=function name, MSG=I/W/E/S/N</i>	temporarily changes the MSG parameter of the specified function to information (I), warning (W), error (E), severe (S), or none (N)
<i>T, FUNC=function name, SMF=E/I/N</i>	temporarily changes the SMF message parameter for the specified function to information (I), warning (W), error (E), severe (S), or none (N)
<i>T, MSGID, YES/NO/DEB</i>	temporarily changes the message ID to allow or suppress the message ID from being issued with MAINVIEW SRM messages <b>Note:</b> Use the debug option (DEB) only as directed by BMC Software Customer Support. This option adds diagnostic information to messages that are written to the job log.
<i>T, MSGPREF, xxx</i>	temporarily changes the message prefix characters
<i>T, SMFID, nnn</i>	temporarily changes the SMFID The new value should be in the range from 129 to 255, or 0 (zero) to disable recording. For example: F SVOS, SVALLOC T SMFID, 255
<i>T, TRKCYL, nnnnn</i>	temporarily changes the default cylinder size (tracks per cylinder)
<i>T, TRKLEN, nnnnnn</i>	temporarily changes the default track size (bytes per track)

**CONFIG, REFRESH Command**

Use the CONFIG, REFRESH console command to cause the configuration component to refresh all configuration information. You should only need to use this command when an error has occurred. The configuration component automatically refreshes data when changes are noted. No corresponding REFRESH command available in the view dialog.

This command has no default value. Only one configuration REFRESH can run at a time. If a REFRESH is attempted while another REFRESH is active, an error message is written to the log and the request is rejected.

The following abbreviations are valid for the CONFIG console command:

- R (REFRESH)
- C (CONFIG)

Command examples:

```
SVOS CONFIG, REFRESH
SVOS C, R
```



## CONFIG, DIAG(DUMP|volser|device address) Command

Use the CONFIG, DIAG console command to cause the configuration component to snap diagnostic data. If the DIAG keyword of the command indicates DUMP, MAINVIEW SRM snaps all current configuration information. If the DIAG keyword contains a value other than DUMP, MAINVIEW SRM snaps diagnostic information related to the specified volume serial. Use this command only at the request of BMC Software Customer Support.

This command has no default value. Only one configuration DIAG can run at a time. If a DIAG is attempted while another DIAG is active, an error message is written to the log and the request is rejected.

The following abbreviations are valid for the CONFIG, DIAG console command:

- D (DIAG)
- C (CONFIG)

Command examples:

```
SVOS CONFIG,DIAG(DUMP)
SVOS C,D(DUMP)
SVOS C,D(WORK01)
SVOS C,DIAG(WORK02)
SVOS C,D(85A0)
```

## Tracking MAINVIEW SRM Activity

Use the audit log facility to track actions that MAINVIEW SRM has taken against resources and to record the actions of other processes. For example, VTOC scan will provide a log record to record when a collection has started and ended. Audit logging data sets are created during installation and the defined data sets are added to the MAINVIEW SRM product by using JCL on the ACTLOGnn DD statements at startup.

When logging is performed, the active log DD is selected at startup by scanning the allocated ACTLOGnn DDs and selecting the DD associated with the most recent data or the first empty DD. When the active log data set space is exhausted, the next empty DD in numerical sequence or the DD with the oldest data will become the active log. Any allocated ACTLOGnn DDs are used and then reused in this cyclic fashion.

The audit logs provide the following information:

- process name
- process performed
- resource used
- message text

MAINVIEW SRM formats the information into a record and writes it to a log data set, which you can examine in an audit log view.

## Audit Log Views

Table 2-10 provides a description of the audit log views.

**Table 2-10      Audit Log Views**

View	Description
Audit Log Data Set View (LOGFILE) (LOGFILED)	<ul style="list-style-type: none"><li>• displays all logs that are allocated to the ACTLOG<math>nn</math> DD statements</li><li>• contents of the view include data set name, DDname, status, and logging start and end times</li></ul>
Audit Log Display View (LOGREC) (LOGRECD)	<ul style="list-style-type: none"><li>• displays logging information</li><li>• reports the record ID number, component, action user ID, and message text data and up to 10 other optional variable-length USER fields</li></ul>

## Audit Log Records

Table 2-11 describes the audit log records that MAINVIEW SRM writes to the audit log data sets.

**Table 2-11      Audit Log Records (Part 1 of 2)**

Record	Description
Automated Resource End	<ul style="list-style-type: none"><li>• written by the AUTO functions whenever automation on a resource ends</li><li>• provides the end result of the automation</li><li>• supplies the view for seeing automation associated with a resource (pool, volume, application)</li></ul>
Automated Resource Start	<ul style="list-style-type: none"><li>• written by the AUTO functions whenever automation on a resource is started</li></ul>
Configuration Discovery	<ul style="list-style-type: none"><li>• written at the end of each configuration collection request</li><li>• includes the CPU and elapsed times of the configuration request, along with counts of resources including number of volumes, number of physical disks, number of groups, number of pools and subpools, and number of EMC devices</li></ul>

**Table 2-11      Audit Log Records (Part 2 of 2)**

<b>Record</b>	<b>Description</b>
Job End	<ul style="list-style-type: none"><li>• written when a submitted job ends</li><li>• includes the job name, reason for submission, and the ID of the system on which the job was executed</li></ul>
Job Submission	<ul style="list-style-type: none"><li>• written for any job submitted through the AUTO functions using the ACT_JOB keyword</li><li>• written for any job submitted through the view TAG facility</li></ul>
StopX37/II Activity	<ul style="list-style-type: none"><li>• written for all messages created by StopX37/II</li></ul>
VTOC Scan End	<ul style="list-style-type: none"><li>• written by VTOC scan whenever a scan ends</li><li>• provides the end result of the scan</li></ul>
VTOC Scan Start	<ul style="list-style-type: none"><li>• written by VTOC scan whenever a scan is started</li></ul>



---

# Chapter 3     Defining the MAINVIEW SRM System

This chapter provides an overview of the global operational characteristics and system-wide functions of MAINVIEW SRM. The following topics are discussed:

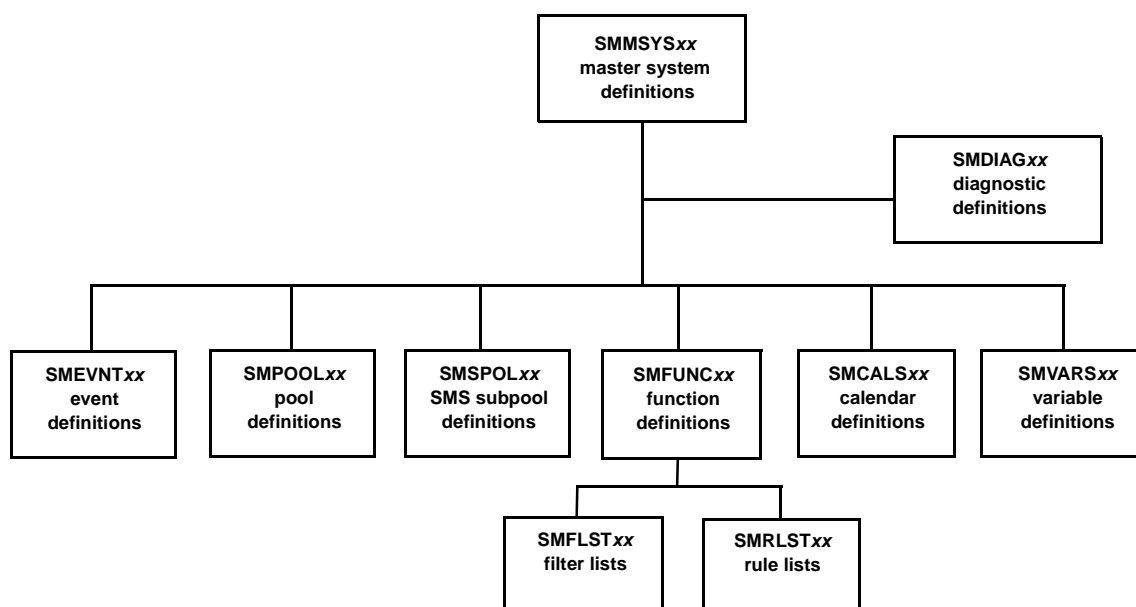
Overview .....	3-2
Defining the System Configuration .....	3-3
MAINVIEW SRM Parameters .....	3-6
Action Parameters .....	3-6
Selection Parameters .....	3-6
Defining the Sysplex Environment .....	3-10
Sysplex Parameters .....	3-11
Syntax Considerations for the Shared Parmlib Environment .....	3-11
Ordering of Selection Criteria .....	3-12
Editing Definition Members .....	3-13

# Overview

You define the MAINVIEW SRM system by coding global partitioned data set (PDS) members and storing them in the parameter library. The global PDS members shown in Figure 3-1 are stored in the MAINVIEW SRM parameter library (BBPARM). They contain the specifications that define the overall characteristics of MAINVIEW SRM. For example, pool members define the composition of DASD pools, and calendar members define company holidays in the calendar year. You specify definitions in text format, and the PDS members interrelate in a hierarchy.

MAINVIEW SRM supports the sysplex environment, which means you can configure your system for a single-system environment or a shared parmlib environment.

**Figure 3-1** MAINVIEW SRM Partitioned Data Set Members



# Defining the System Configuration

A system configuration consists of the SMMSYSxx master system member, which contains global definition parmlib members. Table 3-1, “MAINVIEW SRM Definition Members,” on page 3-4 describes all members and indicates which are global definition members.

A fully functional MAINVIEW SRM system contains at least one configuration of the global definition parmlib members. These parmlib members contain all of the parameters that define a MAINVIEW SRM subsystem.

MAINVIEW SRM reads the parmlib members, beginning with the SMMSYSxx member, during startup to configure the operating environment. SMMSYSxx specifies the suffix of other global definition members that are to be included in the configuration of MAINVIEW SRM that is starting. To define or edit parmlib members, use the procedure described in “Editing Definition Members” on page 3-13.

Multiple MAINVIEW SRM system configurations and multiple suffix versions of the parmlib members, each containing different specifications, can reside in BBPARM. However, only one set of parmlib members is in use by MAINVIEW SRM at any one time.

**Tip:** In a shared parmlib environment, you can implement and maintain MAINVIEW SRM systems by coding statements in a shared OS/390 PDS. For more information, see “Defining the Sysplex Environment” on page 3-10.

- Step 1** Click **Parmlib Members** on the EZSRM Menu. A list of members is displayed.
- Step 2** Select the member from the selection list.
- Step 3** Type **E** in the **CMD** column next to an active member; then press **Enter**.
- Step 4** Edit the member to your specifications.
- Step 5** Press **PF3** to save and exit the member.
- Step 6** Type **R** in the **CMD** column next to the member you edited; then press **Enter**.

The member is refreshed. By refreshing the member, you have made it the active member. It will remain active until SVOS is stopped or until you refresh another member.

Table 3-1 describes each MAINVIEW SRM definition member. The first six characters of each member name are reserved by MAINVIEW SRM; you specify the last two characters.

**Table 3-1 MAINVIEW SRM Definition Members (Part 1 of 2)**

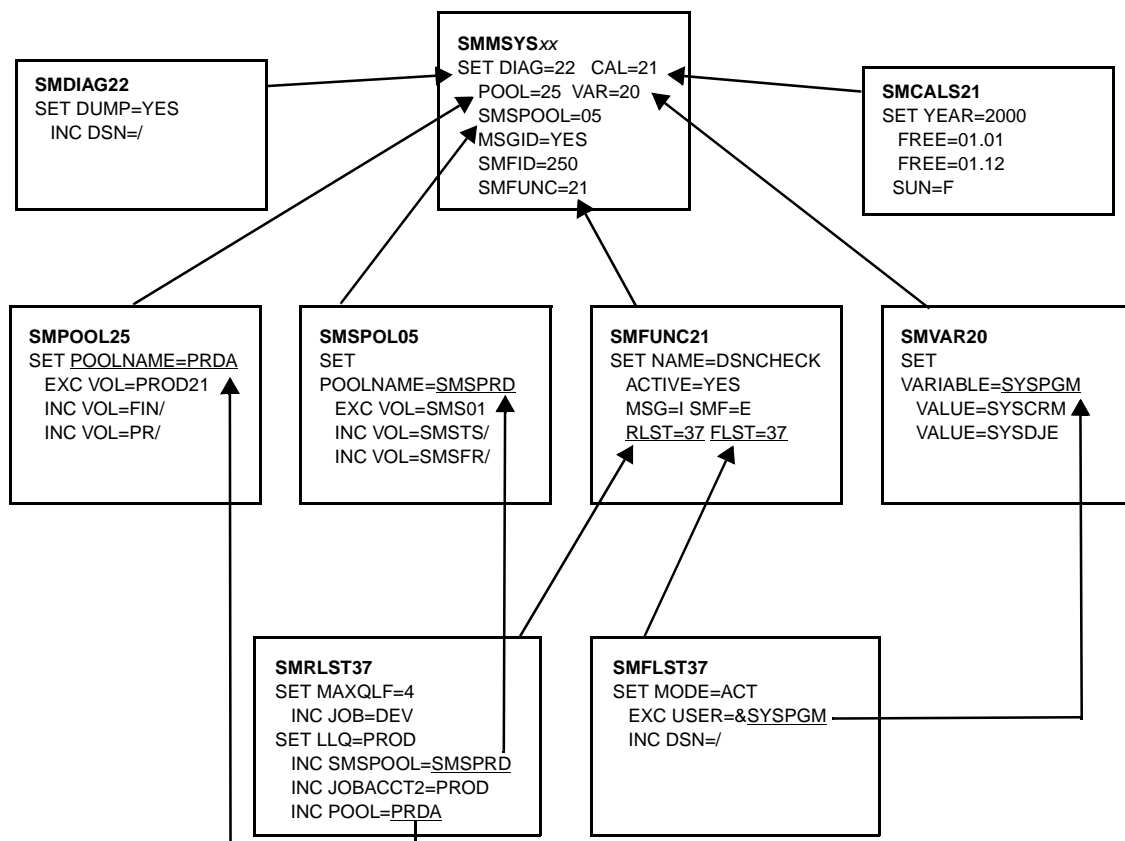
Member Name	Global Definition Member?	Description
SMMSYSxx	No	<ul style="list-style-type: none"> <li>contains <i>all global process options</i> that apply to the overall functionality of MAINVIEW SRM</li> <li>specifies default device type information, message control information, passwords, and SMF IDs</li> <li>during startup, is read before other configuration members</li> <li>specifies the other definition members (see Figure 3-1 on page 3-2) to include in the MAINVIEW SRM configuration that is being started</li> </ul>
SMACMDxx	Yes	<p>specifies matrix and command members that are used for storage automation</p> <p>SG-Auto uses a command member or a combination of a command member and matrix members to perform the functions of monitoring system usage and automating procedures for corrective action. Matrix members are called by command members.</p>
SMCALsxx	Yes	defines nonworking days (weekends, public holidays, company-specific holidays, and so on) to assist in DFHSM migration
SMCRITxx	Yes	enables you to identify a list of critical data sets in your environment that should never be placed on the same volume as any other data set in the list
SMDIAGxx	Yes	specifies various diagnostic options such as tracing of filter and rule list processing or other debugging options that BMC Software Customer Support might request
SMEVNTxx	Yes	defines how storage event notices are managed in Automation Event member parameters include mode, destination, severity, and so on.
SMFLSTxx	No	<p>selects the resources on which a MAINVIEW SRM function operates</p> <p>Resources can be selected by data set name, volume name, unit name, data set type, job name, account number, or various other characteristics. For example, all data sets with names that begin with GL225 or all jobs with a name that ends with TST may be selected. A powerful name-masking facility allows flexibility in identifying the resources to select.</p>
SMFUNCxx	Yes	<p>activates MAINVIEW SRM functions and controls message and tracing activity</p> <p>MAINVIEW SRM space management services are divided into <i>functions</i>. Functions provide all runtime services of MAINVIEW SRM. SMFUNCxx points to members SMFLSTxx and SMRLSTxx, which select resources and control the operation of the functions. With the exception of the USERVARS function, functions and the filter lists and rule lists that are used by a function are specific to individual components.</p>
SMPOOLxx	Yes	defines pools of DASD volumes and pools of tape devices



**Table 3-1** MAINVIEW SRM Definition Members (Part 2 of 2)

Member Name	Global Definition Member?	Description
SMRLSTxx	No	defines rules that control the action of the MAINVIEW SRM function on the selected resources Rule parameters enable you to modify properties such as block size, pool name, and data class. They also enable you to further identify affected resources based on data set name, volume name, job name, data set size, and so on.
SMSPOLxx	Yes	defines subpools of SMS-managed DASD volumes SMS subpools are used only by the Allocation component.
SMVARSxx	Yes	defines MAINVIEW SRM variables and assigns values to those variables MAINVIEW SRM variables can be substituted for MAINVIEW SRM parameter values.

Figure 3-2 shows how the member parameters interconnect to configure a typical system.

**Figure 3-2** Connecting Definition Member Parameters

# MAINVIEW SRM Parameters

Specifications in the parmlib members are in the form of statements that contain parameters. Each parameter is composed of a keyword followed by a value. Parameters are either *action parameters* or *selection parameters*.

## Action Parameters

Action parameters specify values that either control MAINVIEW SRM operation or set a value that is subsequently used by OS/390 or DFSMS.

### Example

```
VOL=( PRGL01 , PRAP11 )
```

tells MAINVIEW SRM which volumes comprise a pool

```
DATACLAS=VSAMLINR
```

sets the DFSMS data class for a new data set

```
EXPDT=96001
```

sets the expiration date for a new data set

```
MSG=E
```

tells MAINVIEW SRM to generate only error messages

You specify rule list parameters on SET statements (statements that begin with the keyword SET). Action parameters can be specified at the system definition level or the function level.

## Selection Parameters

You use selection parameters in filter list and rule list members to define the resources to be selected (see Appendix E“Filter and Rule List Parameters” on page E-1). You also use selection parameters in pool definitions to specify the components of a pool.

You specify selection parameters on INC (include) or EXC (exclude) statements. INC statements select a resource for processing if the resource meets the criteria in the selection parameters. EXC statements exclude a resource from processing if it meets the criteria in the selection parameters.

Selection parameter values are compared to OS/390 properties; if the selection parameter is equal to the OS/390 value, the resource is selected for processing by the MAINVIEW SRM function.

---

**Example**

---

DD=SORTOUT

selects data sets with a DD name of SORTOUT

JOBACCT2=TEST

selects data sets in jobs with a second job account field that contains TEST

MAXSIZE>100MB

selects data sets with a maximum possible size exceeding 100 megabytes

SMSMANAGED=NO

selects data sets that are not under DFSMS control

VOL=PROD23

selects data sets that reside, or will reside, on volume PROD23

---

The flexibility in selecting resources that MAINVIEW SRM provides enables a function to operate on every data set in the data center, on a single data set for a single user, or on any group of data sets.

## Filter Lists

A filter list selects resources for one or more functions and defines the processing mode of those resources to be *active*, *inactive*, or *simulate*. You can use multiple selection criteria in a filter list. The function definition identifies the filter list to be used. A filter list for the DSNCHECK function might look like this:

### Example

```
SET MODE=SIM
INC XMODE=JOB
```

selects all data sets in jobs that are running in batch mode

The DSNCHECK function for these resources is to run in simulate mode.

```
SET MODE=ACT
INC XMODE=TSO
```

selects all data sets in jobs that are running from TSO

The DSNCHECK function for these resources is to run in active mode.

## Rule Lists

A rule list selects resources that are modified with specific values in action parameters. A rule list is invoked for the group of resources selected by the filter list; individual rules in the list can select all or some resources from that group for processing.

You can specify multiple rules in a rule list. The function definition identifies the rule list to use. The following example shows a rule list for the SPACLIMI function.

### Example

```
SET LIMIT=10MB
INC DISP2=KEEP
```

sets an upper size limit of 10 megabytes on data sets with a normal termination disposition of KEEP

```
SET LIMIT=100MB
INC JOBACCT2=PR
```

sets an upper size limit of 100 megabytes on data sets with a second job account field that is PR

## Name Masking

The value you specify for a parameter can be a number (LIMIT=100MB), a literal YES or NO (REPLACE=YES), a single character (MSG=I), or a character field (data set names, volume names, job names, data classes, job account fields, program names, pool names, RACF groups, and so on).

You can specify character-field parameters and name parameters, in full (for example, JOB=PGLYTD25) or you can use *name masking* to specify a partial name (for example JOB=PGL/). Name masking enables the resource-selection parameters to select large or small groups of resources.

### Example

```
INC DSN=HR25.MSTR.YTD
```

selects the single data set HR25.MSTR.YTD

```
INC DSN=* .MSTR.YTD
```

selects all data sets with the final two name qualifiers MSTR.YTD, regardless of the first name qualifier

```
INC DSN=** .MSTR.YTD
```

selects all data sets with the final two name qualifiers MSTR.YTD, regardless of any beginning qualifiers

You cannot use name masking with parameters that require a numeric value or with a member of a predefined set of values.

## Variables

The variables-definition facility further extends the MAINVIEW SRM resource-selection capabilities. A variable can be defined with multiple values and substituted in any selection parameter.

### Example

```
SET VARIABLE=SYSTLIB  
  VALUE=ISP.V*.*LIB  
  VALUE=ISR.V*.*LIB  
  VALUE=SYS?.*LIB
```

defines the variable SYSTLIB to contain the three data set name specifications

in a filter list,

```
INC DSN=&SYSTLIB
```

selects data sets as if these selection parameters were specified:

```
INC DSN=ISP.V*.*LIB  
INC DSN=ISR.V*.*LIB  
INC DSN=SYS?.*LIB
```

## Defining the Sysplex Environment

MAINVIEW SRM uses MAINVIEW infrastructure (MVI) and supports the sysplex environment. The information in this section describes how to define a shared parmlib environment.

In a shared parmlib environment, you can implement and maintain MAINVIEW SRM systems by coding statements in a shared OS/390 PDS. MAINVIEW SRM parmlib members support new INC/EXC parameters for the SET statement and a new capability to override certain SET statements.

You can choose to share SMMSYS<sub>xx</sub> but not SMPOOL<sub>xx</sub> or other members. Sharing parmlib members requires a well-planned naming convention to ensure that pools with different characteristics on different systems do not have the same name.

## Sysplex Parameters

The same parameters that are used in the nonshared environment are supported in the shared environment. To accommodate a shared parmlib environment, the INC/EXC keywords FORSYSID, FORSMFID, and FORPLEXNAME are valid in the following members:

- SMMSYS<sub>xx</sub>
- SMFUNC<sub>xx</sub>
- SMPPOOL<sub>xx</sub>
- SMFLST<sub>xx</sub>
- SMRLST<sub>xx</sub>
- SMDIAG<sub>xx</sub>
- SMEVNT<sub>xx</sub>
- SMCRTIT<sub>xx</sub>
- SMCALS<sub>xx</sub>
- SMSPOL<sub>xx</sub>
- SMVARS<sub>xx</sub>

The SET statement can be overridden, which means that you can code an initial SET statement in the shared SMMSYS<sub>xx</sub>, SMFUNC<sub>xx</sub>, SMPPOOL<sub>xx</sub>, SMDIAG<sub>xx</sub>, SMEVNT<sub>xx</sub>, and SMCRTIT<sub>xx</sub> members, followed by INC/EXC parameters for each image in the sysplex.

The last INC/EXC statement that is coded in a SET statement is accepted as the override value. Previously coded INC/EXC statements are ignored.

## Syntax Considerations for the Shared Parmlib Environment

If no INC FORSYSID is coded on a SET statement, the SET statement applies to all systems in the sysplex.

**Warning!** You *must* code a SET statement after a group of INC/EXC statements if more SET parameters follow.

If multiple SET POOLNAME parameters are found in a shared member for the same POOLNAME, the last value that is coded for the parameter overrides all other values. INC/EXC statements cannot be overridden; that is, they are not allowed on multiple SET statements. This rule allows you to override only the SET parameters, assuming that all volumes in a given pool should be the same on all images in a sysplex. You cannot define pools that contain different volumes on different images with the same name.

**Example**

```
SET POOLNAME=DAVE SGDCollect=NO
INC VOL=BAB/
INC VOL=SHK/
INC SYSID=*
```

specifies that the pool named DAVE *not* collect data for volumes that begin with BAB and SHK on all systems in the sysplex

```
SET POOLNAME=DAVE SGDCollect=YES
INC SYSID=SJSG
```

specifies that the pool named DAVE collects data for the SJSG system only

## Ordering of Selection Criteria

The default value for INC and EXC with the keywords FORSYSID, FORSMFID, and FORPLEXNAME is the current image unless an INC/EXC statement for one of the keywords is coded. In that case, the selection that results from the coded statement prevails.

**Example**

```
INC FORSYSID=SYSG
```

results in the selection of system SYSG only

```
EXC FORSYSID=SYSG
```

results in the selection of all images in the sysplex, except SYSG



## Editing Definition Members

---

**Summary:** Use this procedure to edit and define the MAINVIEW SRM members in your system. Subsequent chapters provide more information about each of the specific members.

---

- Step 1** Click the desired option on the EZSRM Menu to display a list of members. The options are
- **Parmlib members** (see “Editing Definition Members” on page 3-13)
  - **Functions** (see Chapter 4, “Defining the SMFUNCxx Member and Functions”)
  - **Filter List or Rule List** (see Chapter 5, “Using Filter and Rule List Parameters”)
  - **Pools** (see Chapter 6, “Defining a Pool”)
  - **Variables** (see Chapter 7, “Using Variables”)
  - **Calendars** (see Chapter 8, “Defining a Calendar”)
  - **Diagnostics** (see Chapter 9, “Using Diagnostics”)

**Step 2** Select a member from the list.

**Step 3** Type **E** in the **CMD** column next to a member, then press **Enter**.

**Step 4** Edit the member to your specifications.

**Step 5** Press **PF3** to save and exit the member.

**Step 6** Type **R** in the **CMD** column next to the member you edited; then press **Enter**.

The member is refreshed, and thereby made the active member. It remains active until SVOS is stopped or until you refresh another member.



---

## Chapter 4     Defining the SMFUNCxx Member and Functions

This chapter provides instructions for defining and activating functions in MAINVIEW SRM. The following topics are discussed:

Overview .....	4-2
Defining the SMFUNCxx Member .....	4-2
Active and Inactive Functions .....	4-2
SMFUNCxx Member Parameter Reference .....	4-2
Defining and Activating Functions .....	4-4
Defining Parameters for Functions .....	4-4
Function Parameter Reference .....	4-5
Defining the USERVARS Function .....	4-5

## Overview

SMFUNCxx is the function definition parameter member. MAINVIEW SRM reads it during system startup. The suffix that is specified in SMMSYSxx indicates which version of SMFUNCxx the active MAINVIEW SRM configuration is to use.

SMFUNCxx also identifies the filter list and rule list members used by a function. Functions provide all of the MAINVIEW SRM runtime services. SMFUNCxx is a required member, and most MAINVIEW SRM components do nothing without defined function parameters.

## Defining the SMFUNCxx Member

To edit the function definition member in SMFUNCxx, use the procedure “Editing Definition Members” on page 3-13.

## Active and Inactive Functions

A function may be defined in SMFUNCxx but be kept inactive. Functions in the distributed SMFUNCxx member are defined as inactive and must be activated (see “Defining and Activating Functions” on page 4-4). You can remove from the SMFUNCxx member any function definitions that are not required.

**Note:** Filter and rule lists for inactive functions are validated for proper syntax during refresh operations and at system startup.

## SMFUNCxx Member Parameter Reference

Table 4-1 describes the SET statement function parameters that are used in the SMFUNCxx member. See Appendix D, “Global Parameters” for more information.

**Table 4-1** SET Statement Function Parameters (Part 1 of 2)

Parameter	Required	Description
NAME=xxxxxxx	Yes	MAINVIEW SRM-assigned name of the function
FLST=xx	No	suffix of filter list member SMFLSTxx

**Table 4-1 SET Statement Function Parameters (Part 2 of 2)**

Parameter	Required	Description
RLST=xx	No	suffix of rule list member SMRLSTxx
ACTIVE=YES/NO	Yes	status of the function
MSG=I/W/E/S/N	Yes	level of messages to be generated
SMF=I/W/E/S/N	Yes	level of messages to be written to SMF
TRACE=xxxxxxxx	No	jobname of traced MAINVIEW SRM actions
DESC='xxxxxxxxxxxxxxxxxxxx'	No	description of function

## Examples

These examples show how functions are defined in SMFUNCxx.

The following statement defines the DSNCHECK function and makes it *active*, and defines the SPACLIMI function and makes it *inactive*.

SET	NAME=DSNCHECK FLST=56 ACTIVE=YES MSG=I SMF=N DESC='DSN CHECKING'	The MAINVIEW SRM DSNCHECK function is defined and made active. Filter list parameters are in SMFLST56; no rule list parameters are provided. Information and error messages are to be generated. No information or error messages are to go to SMF.
SET	NAME=SPACLIMI ACTIVE=NO FLST=GG RLST=G1 MSG=I SMF=E DESC='SPACE LIMIT' TRACE=TESTJOB	The MAINVIEW SRM SPACLIMI function (space limit) is defined but inactive. Filter list parameters are in SMFLSTGG; rule list parameters are in SMRLSTG1. Information and error messages are generated, but only error messages are sent to SMF. For jobs named TESTJOB, trace messages are generated to detail filter- and rule-list processing.

The following statement shows the relationship between the SMFUNCxx and the SMFLSTxx members.

<b>SMFUNCxx</b> contains	
SET	NAME=SPACLIMI ACTIVE=YES FLST=56 RLST=56 MSG=I SMF=E DESC='SPACE LIMIT'
<b>SMFLST56</b> contains	

SET	MODE=SIM MSG=I INC SIZE>10M	Sets the function to simulation mode for data sets exceeding 10 megabytes.
SET	MODE=ACT MSG=E INC DISP1=NEW LLQ=TEST/	Sets the function active for data sets that have a first disposition parameter of NEW and a low-level data set name qualifier that begins with TEST.

## Defining and Activating Functions

In the SMFUNCxx member that is provided on the distribution tape, all functions are defined as inactive. To use MAINVIEW SRM, you must activate the functions you want to use and define the resources on which the functions act. You can activate the functions all at once or you may activate a single function at a time. To use a function you must first perform the following actions:

- Define a filter list (in an SMFLSTxx member), with MODE=ACTIVE, and with INCLUDE parameters to apply the function to a set of resources (data sets, volumes, pools, jobs, and so on). (See “Using SMFLSTxx—Filter List Parameters” on page 5-2 for instructions on how to define a filter list.)
- Define a rule list (in an SMRLSTxx member) that specifies how resources are affected. (See “Using SMRLSTxx—Rule List Parameters” on page 5-7 for instructions on how to define a rule list.)

## Defining Parameters for Functions

After you define a filter list and a rule list, follow these steps to define the parameters for each function.

- Step 1** Select **Functions** from the EZSRM menu. The ADFUNC view is displayed. ADFUNC lists the status and information for all functions that are defined in the system.
- Step 2** Use any of the following action codes in the **CMD** column next to a specific function name to further define the function:

Action Code	Used to
A	activate the function
BF	browse the filter list
BR	browse the rule list
C	change the filter/rule list suffix and/or the message or SMF level

Action Code	Used to
DF	display the filter list in core storage
DR	display the rule list in core storage
EF	edit the filter list
ER	edit the rule list
I	inactivate the function
R	refresh both the filter and rule list
DR	display the rule list in core storage
RF	refresh the filter list
RR	refresh the rule list
/F	display the contents of storage for the FLST
/R	display the contents of storage for the RLST

**Step 3** Use the refresh action codes as needed to update the function, then press **Enter**.

**Step 4** Press **PF3** to save and exit the function.

## Function Parameter Reference

MAINVIEW SRM function member parameters are described in Appendix D, “Global Parameters” and the *MAINVIEW SRM Reference Summary*.

With the exception of the USERVARS function, functions and the filter lists and rule lists used by a function are specific to particular components (see “Defining the USERVARS Function”). User guides for each MAINVIEW SRM component provide descriptions of other functions and the filter list and rule list parameters that are available for them.

## Defining the USERVARS Function

The USERVARS function enables you to set variables that are not included in the set of variables provided by MAINVIEW SRM. The variables that are defined in USERVARS may be subsequently tested and used in filter lists and rule lists for other functions.

SET statement parameters for the USERVARS function are located in member SMRLSTxx. Table 4-2, “SET Statement Parameters for USERVARS Function,” on page 4-6 describes them.

**Table 4-2 SET Statement Parameters for USERVARS Function**

Parameter	Description
USRCn=xxxxxxx	character field for a user-specified variable (1–8 characters) The value of <i>n</i> can be 1–10 (for example USRC1, USRC2, and so on).
USRNy=nnnnnnnn	numeric field for a user-specified variable The value of <i>y</i> can be 1–10 (for example USRN1, USRN2, and so on). The value of <i>n</i> cannot exceed 214783647.

The following example shows how you can use USERVARS to set variables across functions.

<b>SMFUNCxx</b> member		
SET	NAME=USERVARS ACTIVE=YES MSG=I SMF=I FLST=UV RLST=UV DESC='SET USER VARIABLES'	Defines and activates the function.
<b>SMFLSTUV</b> member		
SET	MODE=ACT INC JOB=TESTUSRV	Selects the job to test the variables in active mode.
<b>SMRLSTUV5</b> member		
SET	USRC1=USERVALU USRN1=100 INC DSN=TEST.USER1	Specifies two variables to be set and tested later.
SET	USRC1=USERVAL2 USRN1=200 INC DSN=TEST.USER2	Specifies two variables to be set and tested later.
<b>SMFLSTDP</b> member		
SET	MODE=ACT INC USERC1=USERVALU EXC USERN1>100	The variables established by the USERVARS function have been set and are used in the filter list for DASDPOOL.



---

# Chapter 5      Using Filter and Rule List Parameters

This chapter provides instructions for defining and using filter and rule list parameters in MAINVIEW SRM. The following topics are discussed:

Overview.....	5-2
Using SMFLSTxx—Filter List Parameters .....	5-2
SET Statement Parameters .....	5-2
INC and EXC Parameters .....	5-6
Using SMRLSTxx—Rule List Parameters.....	5-7
Defining Filter and Rule Lists .....	5-10

## Overview

SMFLST $_{xx}$  is the filter list member for one or more functions. MAINVIEW SRM reads it during system startup. The filter-list suffix that is specified in SMFUNC $_{xx}$  indicates which version of SMFLST $_{xx}$  the active MAINVIEW SRM configuration is to use.

SMFLST $_{xx}$  parameters control the resources that are selected for a function. SMFLST $_{xx}$  is an optional member; however, if an SMFLST $_{xx}$  member is not specified for a function, no resources are selected for that function.

Filter lists are defined on SET statements and INC and EXC statements. INC and EXC statements that follow the SET statement identify the resources for selection. The SET statement parameters apply only to the resources that are selected by the INC and EXC statement parameters that follow them.

## Using SMFLST $_{xx}$ —Filter List Parameters

You can change SMFLST $_{xx}$  parameters by using the **Filter List** and **Rule List** options on the EZSRM Parmlib Members menu (see “Defining Filter and Rule Lists” on page 5-10). Filter and rule list parameters are defined in Appendix E, “Filter and Rule List Parameters” and the *MAINVIEW SRM Reference Summary*.

## SET Statement Parameters

Table 5-1 describes the SET statement filter-list parameters.

**Table 5-1** SET Statement Parameter Quick Reference (Part 1 of 2)

Parameter	Required	Description
EVENTID=xxxxx	No	event identification EVENTID= is a SET parameter allowed in FLST or RLST of any function except USEVARS and the AUTO functions (see ACT_EVENTID= in Appendix E, which is used in AUTO functions). If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.
MODE=ACT/INACT/SIM	Yes	status of the function Simulate mode (SIM) is used only for the HSM collector, EasySMS, EasyPOOL, StopX37/II, and DMS2HSM. If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

**Table 5-1 SET Statement Parameter Quick Reference (Part 2 of 2)**

Parameter	Required	Description
MSG=I/W/E/S/N	No	level of messages to be generated The MSG parameter is used only for the allocation component and the HSM collector.
SMF=I/W/E/S/N	No	level of messages to be generated to SMF The SMF parameter is used only for the allocation component and the HSM collector.

**EVENTID=**

**Purpose:** Specifies the identifier that is assigned to the user event in SMEVNT<sub>xx</sub>.

**Syntax:** EVENTID=*xxxxx*

The variable *xxxxx* represents the 5-character string specified on the EVENTID parameter in SMEVENT<sub>xx</sub>. The value I is reserved for system events and must not be used in the first position of a user-defined EVENTID. With the exception of the letter I, you can use any 5 characters or numbers in any position of the EVENTID. Consider creating a naming convention for events that you define. For examples of usage, see the *MAINVIEW SRM Automation User Guide*.

Events that are distributed with MAINVIEW SRM components begin with AP<sub>xxx</sub>, AA<sub>xxx</sub>, AV<sub>xxx</sub>, and AD<sub>xxx</sub>. BMC Software recommends that you do not use these event IDs.

EVENTID= is a SET parameter allowed in FLST or RLST of any function except USEVARS and the AUTO functions (see ACT\_EVENTID= in Appendix E for use in AUTO functions).

**Note:** If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

**Required:** No

**Default:** None

**MODE=**

**Purpose:** The MODE parameter specifies the status of the function for the resources that are selected by following INC and EXC statements. It is the most significant filter list SET statement parameter because it defines the processing mode for the resources that are selected. The MODE parameter can be set to one of the following values:

- ACT: active, the function acts on the selected resources
- INACT: inactive, the function does nothing for the selected resources

When you specify MODE=INACT, the function does not process any of the resources that are selected.

**Note:** If EVENTID= is used on an FLST SET statement with MODE=INACT, the event is still issued.

- SIM: simulate, the function reports activity as if it were active, but it does not actually perform any action on the selected resources

When you specify MODE=SIM, the action is not applied but a message is issued (depending on the MSG parameter) to indicate the action that would be applied if you specified MODE=ACT.

**Note:** Simulate mode (SIM) is used only for the Allocation component and the HSM collector.

Different sets of selected resources can have different processing modes for a function.

**Syntax:** MODE=[ACT | INACT | SIM]

- ACTive
- INACTive
- SIMulate

**Required:** Yes

**Default:** None

**MSG=**

**Purpose:** The MSG parameter specifies the message-generation option for resources that are selected by the parameters that follow it. Informational and error messages can be produced, or all messages can be suppressed. Note that this option overrides the MSG option on the function definition in the SMFUNCxx member.

**Note:** The MSG parameter is used only for the Allocation component and the HSM collector.

**Syntax:** MSG=[I / W / E / S / N]

I Informational and error messages  
W Warning messages  
E Error messages only  
S Severe messages  
N No messages

**Required:** No

**Default:** The MSG parameter on the function definition in SMFUNCxx.

**SMF=**

**Purpose:** The SMF parameter specifies the SMF message-generation option for resources that are selected by the parameters that follow it. Informational and error messages can be written to the SMF data set, or all messages can be omitted from the SMF data set. Note that this option overrides the SMF option on the function definition in the SMFUNCxx member.

**Note:** The SMF parameter is used only for the Allocation component and the HSM collector.

**Syntax:** MSG=[I / W / E / S / N]

I Informational and error messages  
W Warning messages  
E Error messages only  
S Severe messages  
N No messages

**Required:** No

**Default:** None

## INC and EXC Parameters

The INC and EXC parameters for the filter and rule lists are extensive and are explained in Appendix E, “Filter and Rule List Parameters.”

You can specify multiple INC and EXC parameters. When you specify the parameters on the same line, resources are selected when all parameters are true; that is, a Boolean AND condition applies to the parameter criteria. The resource is selected if the first parameter and the second parameter are true.

### Example

```
SET MODE=ACT  
INC JOB=*TST XMODE=JOB
```

Resources are selected if the job name ends with TST *and* the job is running in batch mode.

When you specify multiple parameters on separate lines, resources are selected when one parameter or the other is true; that is, a Boolean OR condition applies to the parameter criteria. The resource is selected if the first parameter or the second parameter is true.

### Example

```
SET MODE=ACT  
INC DSN=ISR./  
INC DSN=ISP./
```

Resources are selected if the data set name begins with ISR *or* if the data set name begins with ISP.

When a resource has satisfied the selection criteria on an INC statement, the resource is selected for processing and no further examination of the filter list is performed. When a resource has satisfied the criteria on an EXC statement, however, the resource is excluded only from that filter list entry; the resource is then passed to the next SET statement in the filter list for consideration. Excluded resources are considered by each entry in the filter list.

- An INC statement selects a resource and exits the filter list.
- An EXC statement rejects a resource for the current entry only and continues checking the filter list.

You can use MAINVIEW SRM name masking in all parameters for the INC and EXC statements. However, you cannot use name masking on SET parameters because a SET statement parameter must specify a definite value. See the information in Appendix A, “MAINVIEW SRM General Syntax Rules,” for full details on selection parameter syntax, including name masking and ordering of statements.

## Using SMRLSTxx—Rule List Parameters

SMRLSTxx is the rule list parmlib member for one or more functions. MAINVIEW SRM reads it during system startup. The rule-list suffix that is specified in SMFUNCxx indicates which version of SMFLSTxx the active MAINVIEW SRM configuration is to use.

SMRLSTxx is an optional member. If an SMRLSTxx member is not specified for a function, the default for the function is used. If there is no default for the function and an SMRLSTxx member is not specified, the function is not processed.

SMRLSTxx parameters control the actions of a function against the storage management objects or properties that are selected for a function. SMRLSTxx parameters also control how MAINVIEW SRM responds to a request from the operating system or DFSMS. For example, an SMRLSTxx member may modify the blocksize or retention period for a data set, or it may reject a volume for data set allocation.

The following functions do not use a rule list because they perform processing that does not need to be tailored for specific resources.

- FORCECAT
- HSMDELET
- OPENEMPT
- SUPJSCAT
- FRAGCNTL
- SUPVOLRF
- MODDELET
- SMSMCREN
- TAPEDEFR

A rule list is composed of one or more rules. A rule is composed of a single SET statement and one or more INC and EXC statements. The SET statement identifies a data or system property to be modified and specifies a value for it.

For example, the following SET statement (for the OPTBLKSZ) function specifies that the block size for new data sets is 10,796:

```
SET  BLKSIZE=10796                (OPTBLKSZ function)
```

In the next example, the SET statement (for the SETEXPT function) specifies an expiration date of (EXPDT) 94180, regardless of any JCL specification (REPLACE), for new data sets:

```
SET  EXPDT=94180 REPLACE=YES      (SETEXPT function)
```

In the next example, the SET statement (for DSNCHECK function) specifies that the maximum number of qualifiers in a data set name is six.

```
SET  MAXQLF=6                    (DSNCHECK function)
```

The INC and EXC statements in rule lists operate in the same way as INC and EXC statements in filter lists to select the specific resources to modify with the SET statement parameter. In the next example, the INC statement (for the OPTBLKSZ function) specifies a mandatory block size for new data sets that are larger than 10 megabytes.

```
SET  BLKSIZE=10796                (OPTBLKSZ function)  
INC  MAXSIZE>10MB
```

In the next example, the INC statement (for SETEXPDT function) specifies that new data sets in any of the test pools (POOL) or with a second job account field (JOBACCT2) of TEST are assigned an expiration date of 94180.

```
SET  EXPDT=94180                (SETEXPT function)  
INC  POOL=TEST/  
INC  JOBACCT2=TEST
```

At least one INC and EXC selection parameter must be specified for a rule. A SET statement without an INC and EXC statement is not applied to any resources.

Selecting resources for a rule is a two-step process. First, a resource is selected by the function's filter list. Next, the resource is passed to the rule list for further checking. A resource must be included in both the filter list and rule list results for the function SET parameters to be applied.



When a resource has satisfied the selection criteria on an INC statement, the resource is selected for processing and no further examination of the rule list is performed. When a resource has satisfied the criteria on an EXC statement, however, the resource is excluded only from that rule list entry; the resource is then passed to the next SET statement in the rule list for consideration. Excluded resources are considered by each entry in the rule list.

- An INC statement selects a resource and exits the rule list.
- An EXC statement rejects a resource for the current entry only and continues checking the rule list.

You can use MAINVIEW SRM name masking in all parameters for the INC and EXC statements. However, you cannot use name masking on rule SET parameters because a SET statement parameter must specify a definite value. See the information in Appendix A, “MAINVIEW SRM General Syntax Rules,” for full details on selection parameter syntax, including name masking and ordering of statements.

### Parameter Explanations

The parameters for the rule and filter lists are extensive, and are explained in Appendix E, “Filter and Rule List Parameters.”

The following example shows how pools are selected and controlled by using the filter and rule lists.

<b>SMFLSTxx</b> contains		
SET	MODE=SIM MSG=E INC POOL=TEST/	Select all data sets that reside in the test pool.
<b>SMRLSTxx</b> contains		
SET	LIMIT=5M INC VOL=TEST5/	All data sets that reside in any test pool are selected by the filter list. The two rules limit these data sets to 5MB on volumes TEST5A and TEST5B and to 2MB on other volumes beginning with TEST.
SET	LIMIT=2M INC VOL=TEST/	

## Defining Filter and Rule Lists

Before you can use a function, you must first perform the following actions:

- Define a filter list (in an SMFLSTxx member), with MODE=ACTIVE, and with INCLUDE parameters to apply the function to a set of resources (data sets, volumes, pools, jobs, and so forth).
- Define a rule list (in an SMRLSTxx member) that specifies how resources are affected.

Use the **Filter List** and **Rule List** menu options to define the parameters in SMFLSTxx and SMRLSTxx. For instructions, see “Editing Definition Members” on page 3-13.

---

# Chapter 6     Defining a Pool

This chapter provides instructions for defining pools in MAINVIEW SRM.  
The following topics are discussed:

Overview .....	6-2
Using the SMPPOOLxx Member .....	6-2
Defining Pools .....	6-5

## Overview

SMPOOLxx is the member in which you define pool parameters. MAINVIEW SRM reads it during system startup. The suffix that is specified in SMMSYSxx indicates which version of SMPOOLxx that the active MAINVIEW SRM configuration is to use. SMPOOLxx is a required member, even in installations where pooling is not used.

**Note:** SMS subpools are used only by the Allocation component (EasyPOOL). For SMS subpooling, use the SMSPOLxx member (see the *MAINVIEW SRM EasyPOOL User Guide and Reference*).

Pool members define the composition of DASD pools. Pools must be defined before you begin using the MAINVIEW SRM Allocation component.

## Using the SMPOOLxx Member

Pools are defined on SET statements. INC and EXC statements that follow a SET statement identify the devices to be included in or excluded from a pool.

**Note:** Offline devices are not placed in pools, although they may be referenced on an INC statement. Offline devices that are assigned to a pool by an INC statement and are later varied online are automatically picked up by MAINVIEW SRM.

You can change SMPOOLxx parameters through the **Pools** option on the EZSRMP Parmlib Members menu (see “Defining Pools” on page 6-5).

## Parameter Quick Reference

Table 6-1, “SET Statement Pool Parameter Quick Reference,” on page 6-3 provides a quick reference to the SET statement parameters for pools. See Appendix D, “Global Parameters” or the *MAINVIEW SRM Reference Summary* for more information.

**Table 6-1 SET Statement Pool Parameter Quick Reference**

Parameter	Required	Description
POOLNAME=xxxxxxx	Yes	name to be assigned to a pool
USELIMIT=nnn	No	upper space threshold for new allocations
SGDCollect=YES/NO	No	determines whether a pool is processed
TYPE=xxxx	No	device type

Table 6-2 provides a quick reference for INC and EXC statement parameters for pools.

**Table 6-2 INC and EXC Statement Pool Parameter Quick Reference**

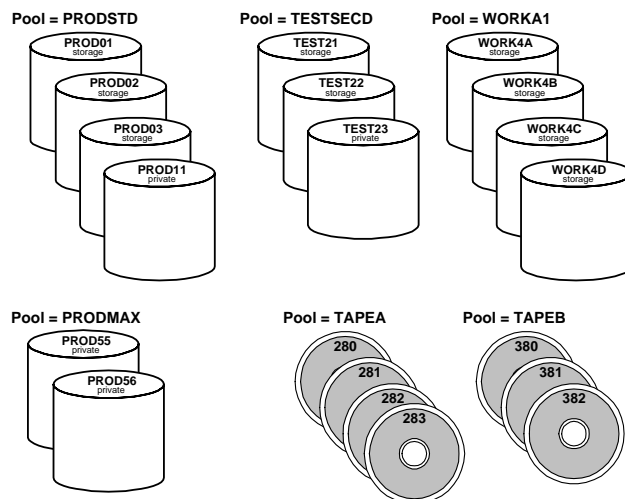
Parameter	Required	Description
ADR=xxxx	No	device address of tape unit in the pool
ADR=(xxxx,xxxx,...)	No	multiple tape device addresses (up to 15)
VOL=xxxxxx	No	volume serial number of the device in the pool
VOL=(xxxxxx,xxxxxx,...)	No	multiple volume serial numbers (up to 15)

**Note:** The MAINVIEW SRM Reporting component supports a volume being defined in up to eight pools and subpools combined. The **Vols Dropped** column on view GPCNFG indicates the number of volumes that have been dropped from pool- or subpool-report information. Configuration information is *not* correct for pools and subpools from which volumes have been dropped.

The MAINVIEW SRM Allocation component supports a volume being defined in an infinite number pools and subpools; therefore, it is not affected by the **Vols Dropped** count.

Figure 6-1 shows an example of a system storage configuration for pools. The statements following the figure define the configuration.

**Figure 6-1 Example Storage Configuration for Pools**



SET POOLNAME=PRODSTD USELIMIT=90 INC VOL=PROD0/ INC VOL=PROD11	The PRODSTD pool includes all volumes with names that begin with PROD0 and volume PROD11. The upper space limit (USELIMIT) is 90%; 10% is generally reserved for expansion.
SET POOLNAME=PRODMAX USELIMIT=80 INC VOL=(PROD55,PROD56)	The PRODMAX pool includes volumes PROD55 and PROD56. The upper space limit is 80% (this pool is for very large data sets, which also need larger secondary extents; 20% is reserved for expansion.)
SET POOLNAME=TESTSECD INC VOL=(TEST21,TEST22,TEST23)	The TESTSECD pool includes volumes TEST21, TEST22, and TEST23. There is no provision for an upper space limit (no USELIMIT is set).
SET POOLNAME=WORKA1 USELIMIT=95 INC VOL=WORK4/	The WORKA1 pool includes all volumes with names that begin with WORK4. The upper space limit is 95%; 5% is generally reserved for expansion.
SET POOLNAME=TAPEA TYPE=3490 INC ADR=(0280,0281,0282,0283)	The tape pool TAPEA includes the 3490 tape units 280-283.
SET POOLNAME=TAPEB TYPE=3480 INC ADR=038%	The tape pool TAPEB includes 3480 tape units 380, 381, and 382.

The following examples show how you can use the various pooling parameters.

SET POOLNAME=&STORGRP SGDCOLLECT=YES INC VOL=/ 	Defines a pool for DFSMS storage groups to include only DFSMS volumes. Turns on the space collector.
SET POOLNAME=WORK SGDCOLLECT=YES INC VOL=WRK/	Uses name masking to define a pool for work volumes to include all volumes with the WRK prefix. Turns on the space collector.

SET POOLNAME=LARG SGDCOLLECT=YES INC VOL=(VOL001,VOL002,VOL005)	Defines a pool for large volumes to include specific volume serial numbers. Turns on the space collector.
SET POOLNAME=&VOLSER(1,3) SGDCOLLECT=YES INC VOL=	Defines a generic pool for unidentified volumes so that the space collector can collect data on them for reporting purposes only.
SET POOLNAME=SORT SGDCOLLECT=YES INC STORGRP=TEMP	INC STORGRP= is valid only for the space collector. If you want to set your own pool names or combine storage groups into a pool, you can code definitions like the ones shown in these examples.
SET POOLNAME=VARIOUS SGDCOLLECT=YES INC STORGRP=(SPILL,PDSE,PROFILE,BUNDL,HFS)	
SET POOLNAME=SYSTEM SGDCOLLECT=YES INC VOL=(MV/,NUS/) INC STORGRP=SYSBASE	
SET POOLNAME=TTAPE INC UNIT=TAPE	Defines tape pools.
SET POOLNAME=TCART INC UNIT=CART	
SET POOLNAME=TTAPE2 INC ADR=(0480, 0481, 0482)	

## Defining Pools

After you have identified the criteria for pools at your data center, you can begin defining pools. Use the **Pools** menu option to define the parameters in SMPOOLxx. For instructions, see “Editing Definition Members” on page 3-13.

**Note:** Certain pools created in SMPOOLxx are supported only by the space collector. These space collector pools are any that use the following specifications in the pool definition:

```
SET POOLNAME=&VOL
SET POOLNAME=&UNIT
SET POOLNAME=&MNTYP
```

These pools appear only under the Space Collector Pools option of the Space Utilization menu.





---

# Chapter 7      Using Variables

This chapter provides instructions for defining and using variables in MAINVIEW SRM. The following topics are discussed:

Overview .....	7-2
Using the SMVARSxx Member .....	7-2
Usage Notes .....	7-3
Examples .....	7-4
Defining the Variables Option .....	7-5

## Overview

SMVARS<sub>xx</sub> is the variables definition parameter member. MAINVIEW SRM reads it during system startup. The suffix that is specified in SMMSYS<sub>xx</sub> indicates which version of SMVARS<sub>xx</sub> the active MAINVIEW SRM configuration is to use.

SMVARS<sub>xx</sub> variables allow a user-specified name to be assigned any number of selection parameters. The variable name is then specified in any other MAINVIEW SRM selection statement.

You can change SMVARS<sub>xx</sub> parameters through the **Variables** option on the EZSRM Parmlib Members menu (see “Defining the Variables Option” on page 7-5). SMVARS<sub>xx</sub> is an optional member.

## Using the SMVARS<sub>xx</sub> Member

Table 7-1 provides a quick reference to the SET statement variable parameter. See Appendix D, “Global Parameters” or the *MAINVIEW SRM Reference Summary* for more information.

**Table 7-1 SET Statement Variable Parameter**

Parameter	Required	Description
VARIABLE=xxxxxxxxx	Yes	Specifies the name that is assigned to the variable.

Table 7-2 provides a quick reference to the INC and EXC statement variable parameter.

**Table 7-2 INC and EXC Statement Variable Parameter**

Parameter	Required	Description
VALUE=xxxxxxxxxxxxx	Yes	Specifies a value for the variable.

## Usage Notes

Variables are specified by SET statements. One or more VALUE statements following the SET statement identify the values of the variable. Any selection parameter values that are valid for filter and rule lists can be specified in the values of the variable. For example:

---

```
SET VARIABLE=TESTDSN
    VALUE=TEST/
    VALUE=** .TEST
```

---

At execution time, variables are substituted for selection-statement parameter values. For example:

---

```
INC DSN=&TESTDSN
INC DSN=&TESTPREFIX&DIVCODE&FILETYPE
```

---

**Note:** Blanks or commas are not allowed between concatenated variable names.

See “Name Masking” on page A-6 for more information on the use of variables in selection and action parameters.

Variables are supported for non-Boolean text fields only. Variable values should not include any parameter keywords, other variable names, embedded blanks, or the INC and EXC operands. For example, the following statements are *invalid*:

---

```
VALUE=INC DSN=TEST/    (invalid use of INC operand)
VALUE=TEST/  PROD/    (invalid use of an embedded blank)
VALUE=&APMSTR          (invalid use of a variable name)
VALUE=' POOL=PROD*'    (invalid use of a parameter keyword)
```

---

## Examples

The following examples show how to define and use variables.

### Definition

```
SET VARIABLE=FINCVOL  
VALUE=FINC/  
VALUE=GLAP/  
VALUE=PAY/  
VALUE=RECV9?
```

FINCVOL is a variable that is defined with four volume specifications.

### Usage

```
INC VOL=&FINCVOL
```

### Definition

```
SET VARIABLE=TESTDSN  
VALUE=**.TEST*  
VALUE=*.*TSTMSTR  
VALUE=CICSTEST./
```

TESTDSN is a variable that is designed to include any data set names that have final qualifier that begins with TEST, or that the third qualifier is TSTMSTR, or with a high-level qualifier that is CICSTEST.

### Usage

```
SMFLSTxx SET MODE=ACT  
INC DSN=&TESTDSN
```

A variable is an excellent method of defining such widely used specifications as job classes or programmer groups; for example:

---

```
SET VARIABLE=TESTJOBS  
VALUE=T  
VALUE=S  
VALUE=K  
SET VARIABLE=PRODJOBS  
VALUE=P  
VALUE=A  
VALUE=C  
VALUE=U  
SET VARIABLE=STGADM  
VALUE=JSMITH/  
VALUE=STGADM/  
VALUE=TMALLEY/
```

---

The &TESTJOBS and &PRODJOB variables can be used in any selection criteria to distinguish between test and production jobs. The &STGADM variable can be used in any selection criteria to include or exclude resources for the storage administrator; if the storage administrator changes, only the variable definition must be changed, rather than all the filter and rule lists that are based on the storage administrator's user ID.

## Defining the Variables Option

After you have identified the criteria for variables at your data center, use the **Variables** menu option to define variables. For instructions, see “Editing Definition Members” on page 3-13.



---

# Chapter 8     Defining a Calendar

This chapter provides instructions for defining and using the calendar in MAINVIEW SRM. The following topics are discussed:

Overview . . . . .	8-2
Using the SMCALSxx Member . . . . .	8-2
Defining the Calendars Option . . . . .	8-3

## Overview

Calendar members define company holidays and other non-working days in the calendar year. SMCALSxx is the calendar definition parameter member. MAINVIEW SRM reads it during system startup. The suffix that is specified in SMMSYSxx indicates which version of SMCALSxx the active MAINVIEW SRM configuration is to use. SMCALSxx is an optional member.

You use SET statements to define calendar years. INC statements following the SET statement identify the non-working days to be included. SMCALSxx parameters identify non-working days that are not to be considered as usage days by DFHSM. SMCALSxx offers much more flexibility in defining non-working days than does DFHSM.

**Note:** Any day not specified in SMCALSxx is a working day.

You can change SMCALSxx parameters through the **Calendars** option on the EZSRMP Parmlib Members menu (see “Defining the Calendars Option” on page 8-3).

## Using the SMCALSxx Member

Table 8-1 provides a quick reference to the SET statement calendar parameter. See Appendix D, “Global Parameters” or the *MAINVIEW SRM Reference Summary* for more information.

**Table 8-1 SET Statement Calendar Parameter**

Parameter	Required	Description
YEAR=nnnn	Yes	year to which the following days apply

Table 8-2 provides a quick reference to the INC and EXC statement calendar parameters.

**Table 8-2 INC and EXC Statement Calendar Parameters**

Parameter	Required	Description
FREE=nn.nn-nn.nn	No	from-to range of non-working (free) days
MON=F/W	No	day of week
TUE=F/W	No	day of week
WED=F/W	No	day of week
THU=F/W	No	day of week



**Table 8-2 INC and EXC Statement Calendar Parameters**

Parameter	Required	Description
FRI= <i>F/W</i>	No	day of week
SAT= <i>F/W</i>	No	day of week
SUN= <i>F/W</i>	No	day of week

**Example**

The following statement examples show how you can use various calendar parameters.

SET	YEAR=2001 SUN=F FREE=01.05 FREE=12.06 FREE=24.11-30.11	Non-working days in 2001 include: all Sundays, May 1, June 12, and November 24–30. Note that dates are specified day, and then month.
SET	YEAR=2002 FREE=02.01 FREE=13.10 FREE=21.12-26.12	Non-working days in 2002 are January 2, October 13, and December 21–26

## Defining the Calendars Option

After you have identified the criteria for variables at your data center, use the **Calendars** menu option to define calendars. For instructions, see “Editing Definition Members” on page 3-13.



---

# Chapter 9      Using Diagnostics

This chapter provides instructions for when and how to use diagnostics in MAINVIEW SRM. The following topics are discussed:

Passive Diagnostics . . . . .	9-2
Active Diagnostics . . . . .	9-2
Using the SMDIAGxx Member . . . . .	9-3
Defining the Diagnostics Option . . . . .	9-5
Using the Tracing Facility. . . . .	9-5
Activating Tracing . . . . .	9-5
How Function Tracing Works . . . . .	9-7
Additional Trace Options . . . . .	9-10

## Passive Diagnostics

This release of MAINVIEW SRM provides a new diagnostic capability in the configuration-discovery code. This passive diagnostic capability performs these actions:

- When MAINVIEW SRM detects an error, it snaps data to the SNAP DD statement. It snaps all data from low-level data sources, including all channel command output, EMC API output, and other source output.
- When MAINVIEW SRM detects an abend, it snaps the SVOS address space to the SNAP DD. During the life of an SVOS instance, only two full address space snaps will be produced to prevent overuse of the SNAP DD.

## Active Diagnostics

Diagnostic parameters are available for use when you are directed by BMC Software Customer Support to use them. The remainder of this chapter provides an explanation of how the SMDIAG $_{xx}$  system parameter is used and provides an explanation and example of the tracing facility.

**Warning!** MAINVIEW SRM logic has internal ESTAEs designed to capture SDUMP data for in-depth diagnostics. If you are using another diagnostic product, such as Compuware Abend-AID, that also captures dump information, you should disable the other diagnostic product for MAINVIEW SRM products. Consult your software vendor for the appropriate method of deactivating the diagnostic product.

SMDIAG $_{xx}$  is the diagnostic parameter member.

MAINVIEW SRM reads it during system startup. The DIAG= $_{xx}$  suffix that is specified in SMMSYS $_{xx}$  indicates which version of SMDIAG $_{xx}$  contains diagnostic specifications that the active MAINVIEW SRM configuration is to use.

## Using the SMDIAGxx Member

Diagnostic definitions are specified by SET statements. INC and EXC statements following the SET statement identify the specific situations to be debugged.

Table 9-1 describes the valid SET and INC statements available in the SMDIAGxx member.

**Warning!** Use only the valid combinations listed or unpredictable results can occur.

**Table 9-1 SMDIAGxx SET and INC Statements**

SET Variable	INC Variable				
	FUNCTION	JOB	STEP	PGM	MODULE
TRACE	Y	Y	Y	Y	Y
DEBUG	N	Y	Y	Y	Y
MODTRC	N	Y	Y	Y	Y
DUMP	N	Y	Y	Y	Y
ABEND	N	N	N	N	Y
IGNORE	N	Y	Y	Y	N

FUNCTION, JOB, STEP, and PGM variables are not valid with the ABEND action parameter. The MODULE variable is not valid with the IGNORE action parameter. The FUNCTION variable is not valid with the MODULE action parameter.

The TRACE action parameter (see “Using the Tracing Facility” on page 9-5) can be used instead of //PROTRACE to get an FLST/RLST trace; otherwise, SMDIAGxx should be used only at the direction of BMC Software technical support. The action parameters specified in the SMDIAGxx member override any special JCL DD statements, such as //PROTRACE, as well as the TRACE function parameter.

The IGNORE parameter will override the // PROIGN DD statement. If the DIAG member specifies IGNORE=NO, even the presence of a //PROIGN DD statement will not prevent MAINVIEW SRM functions from being activated.

## Parameter Quick Reference

Table 9-2 provides a quick reference to the SET statement diagnostics parameters. See Appendix D, “Global Parameters” or the *MAINVIEW SRM Reference Summary* for more information.

**Table 9-2 SET Statement Diagnostics Parameters**

Parameter	Required	Description
ABEND=YES/NO	No	forces S0C3 abend when a particular module is entered
DEBUG=YES/NO	No	optional debugging information from a MAINVIEW SRM module
DUMP=YES/NO	No	issues SDUMP if program abend occurs
IGNORE=YES/NO	No	skip this function
MODTRC=YES/NO	No	MAINVIEW SRM module trace
TRACE=YES/NO	No	FLST/RLST trace output

Table 9-3 provides a quick reference to the INC and EXC statement diagnostics parameters.

**Table 9-3 INC and EXC Statement Diagnostic Parameters**

Parameter	Description
FUNCTION=xxxxxxx	valid MAINVIEW SRM function name (up to eight characters)
JOB=xxxxxxx	job name (up to eight characters)
MODULE=xxxxxxx	valid MAINVIEW SRM module name (up to eight characters)
PGM=xxxxxxx	valid MAINVIEW SRM program name (up to eight characters)
STEP=xxxxxxx	step name (up to eight characters)

This example shows how you can use diagnostic parameters.

SET TRACE=YES DEBUG=NO MODTRC=NO DUMP=NO INC JOB=RICKM/ FUNCTION=SPACLIMI	sets the trace parameter for the SPACLIMI function and job name RICKM
SET IGNORE=YES TRACE=NO DEBUG=NO MODTRC=NO X DUMP=NO INC FUNCTION=OPTBLKSZ JOB=JERRY/	skips the OPTBLKSZ function on job name JERRY

## Defining the Diagnostics Option

After you have identified the criteria for diagnostics at your data center, use the **Diagnostics** menu option on the EZSRMP Parmlib Members menu (see “Defining the Diagnostics Option” on page 9-5) to define diagnostics. For instructions, see “Editing Definition Members” on page 3-13.

## Using the Tracing Facility

You can display the MAINVIEW SRM selection processing steps with the MAINVIEW SRM trace facility. The trace output shows the exact steps a MAINVIEW SRM function takes in selecting resources and applying rule-list operands.

## Activating Tracing

The most flexible mechanism for tracing MAINVIEW SRM function activity is to use the SMDIAG<sub>xx</sub> member of the MAINVIEW SRM parmlib. This mechanism enables you to specify tracing by job name, function, and so on and gives you full masking capability on each of the selection fields. You can turn tracing on or off dynamically without the need to make JCL modifications.

Alternatively, you can declare tracing at the individual function level for a specific job. All MAINVIEW SRM activity for that function is traced for any job with the specified name. The TRACE action parameter on the function definition SET statement specifies the name of the target job:

---

```
SET  FUNCTION=DSNCHECK
      ACTIVE=YES
      TRACE=PR44LGX           trace job PR44LGX for DSNCHECK
      MSG=I
      FLST= . . . . .
```

---

Note that the job name that you specify on the TRACE parameter must be exact; name masking does not apply. The use of the SMDIAG<sub>xx</sub> parmlib member makes tracing far more flexible than function tracing in previous releases of MAINVIEW SRM.

You can also activate tracing by specifying `TRACEDD=ddname` in the `SMMSYSxx` parmlib member and including a DD name in the JCL for the job to be traced. All MAINVIEW SRM functions are traced for the job step that contains the specified DD name. (See “Additional Trace Options” on page 9-10.)

The output of the trace is a series of MAINVIEW SRM messages (SMS0042I), which are directed to the job log or TSO user. The following variations of the 042I message are issued:

- selection start

This message is issued at each `SET` statement (the beginning of a selection process) and is identified by a question mark as the first character in the text:

```
SMS0042I TRACE ?SET FLST
```

The message text identifies a `SET` statement and the source (filter list or rule list) of the statement.

- selection failure

This message is issued for each parameter of each `INC` or `EXC` statement that *is not satisfied* by the current resource. It is identified by a minus sign as the first character in the text:

```
SMS0042I TRACE - JOB : EMPCRMX = EMPDJE/
```

The message text identifies the selection parameter (`JOB`, in this example), the current resource name (`EMPCRMX`), and the parameter value (`EMPDJE/`).

- selection success

This message is issued for each parameter of each `INC` or `EXC` statement that *is satisfied* by the current resource. It is identified by a plus sign as the first character in the text:

```
SMS0042I TRACE + JOB : EMPCRMX = EMPC/
```

The message text identifies the selection parameter (`JOB`, in this example), the current resource name (`EMPCRMX`), and the parameter value (`EMPC/`).



## How Function Tracing Works

The example in Figure 9-1 shows how function tracing works. In the example, the job EMPCRMX is defined in SMFUNCxx as the job to be traced. This particular job allocates a new data set that is named EMPCRM.SPACLIMI.TEST.

**Figure 9-1      Tracing Example**

---

```
//EMPCRMX   JOB  (3500) , 'MORRIS' ,MSGCLASS=R ,CLASS=F
//STEP10    EXEC  PGM=IEFBR14
//DD2       DD   DSN=EMPCRM.SPACLIMI.TEST ,
//           DISP=(NEW,DELETE,DELETE) ,
//           SPACE=(TRK,(100,100)) ,UNIT=BABDA ,
//           DCB=(LRECL=80 ,BLKSIZE=3120 ,RECFM=FB ,DSORG=PO)
```

---

One-hundred tracks are requested for the new data set; the limit in the rule list is 1 kilobyte. The filter list and rule lists are shown in Figure 9-2, “Tracing a Function” on page 9-9. The numbers in the following explanation correspond to the numbers in the figure:

1. The trace is activated for the SPACLIMI function when job name EMPCRMX is found to match the job name selection parameter on the INC statement.
2. The first SET statement in the filter list is checked (?SET FLST).

The first selection statement fails (SMS0042I TRACE - JOB ...message is issued) because the resource job name EMPCRMX does not match the name mask EMPDJE/.

The second selection statement is checked because it has an OR relationship with the first statement. The selection fails (- DSN ...), because the resource data set name EMPCRM.SPACLIMI.TEST does not match the name mask EMPDJE/.

3. The second filter list SET statement is checked (?SET FLST).

The first selection statement fails (SMS0042I TRACE - JOB ...) because the resource job name EMPCRMX does not match the name mask EMPEJD/.

The second selection statement fails (SMS0042I TRACE - JOB ...) because the resource job name EMPCRMX does not match the name mask EMPJBS/. The second selection parameter in the INC statement is not checked, because it has an AND relationship with the first parameter, which failed.

The third selection statement succeeds (SMS0042I TRACE + JOB ...) because the resource job name EMPCRMX matches the name mask EMPCRM/.

Because the XMODE parameter is on the same line as the JOB parameter, the two conditions have an AND relationship; therefore, the XMODE parameter is also checked. The selection succeeds because the resource-job execution mode (batch) matches the XMODE parameter value of JOB.

4. The filter list check is complete because the resource has been selected. The last SET statement in the filter list is not checked.

The trace shows a successful filter list selection (SMS0042I TRACE +INC FLST) and a successful filter list execution (SMS0042I TRACE +SET FLST).

5. The rule list for the SPACLIMI function is now checked (SMS0042I TRACE ?SET RLST).

The selection statement succeeds (SMS0042I TRACE + DSN ...) because the resource data set name EMPCRM.SPACLIMI.TEST matches the name mask EMPCRM.\*.TEST.

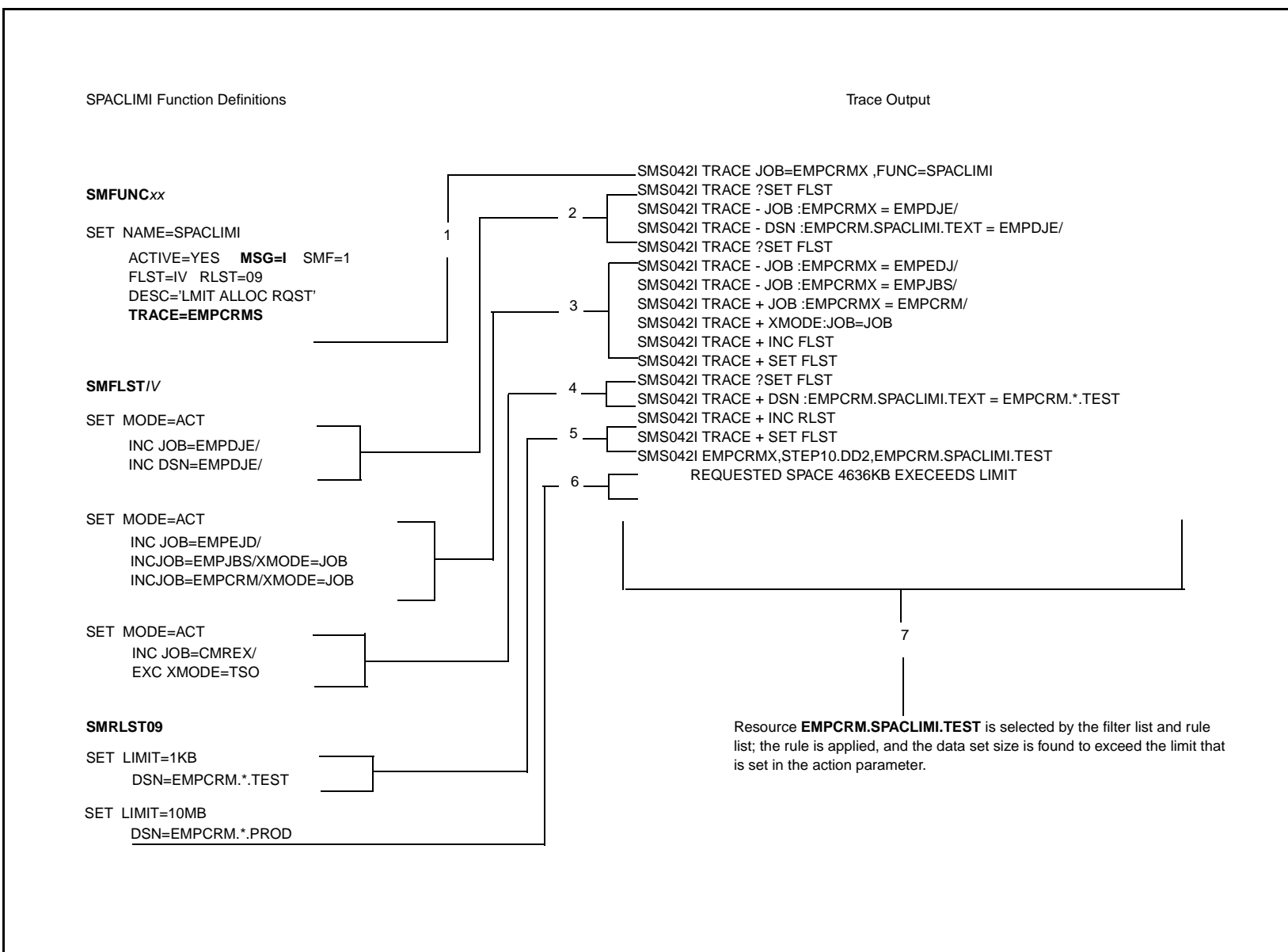
6. The rule list checking is complete because the resource has been selected. The final SET statement in the rule list is not processed.

The limit of 1KB is enforced for this resource.

The trace shows a successful rule list selection (SMS0042I TRACE +INC RLST) and a successful rule list execution (SMS0042I TRACE +SET RLST).

7. The SPACLIMI function causes the data set allocation to fail because the requested space for the data set (100 tracks) exceeds the limit of 1KB in the rule list.

Figure 9-2 Tracing a Function



## Additional Trace Options

You can include the following parameters in the SMMSYSxx parmlib member to control tracing activity:

Parameter	Default	Description
MODTRCDD= <i>ddname</i>	PROMTRCE	activates MAINVIEW SRM module entry/exit tracing
TRACEDD= <i>ddname</i>	PROTRACE	activates standard MAINVIEW SRM tracing (as in TRACE=job name)
IGNOREDD= <i>ddname</i>	PROIGN	deactivates MAINVIEW SRM for duration of job step
DUMPDD= <i>ddname</i>	PRODUMP	produces a SYS1.DUMPxx dump if MAINVIEW SRM abends
DIAGMSDD= <i>ddname</i>	PRO\$D\$M\$	activates WTO message tracing

If you include in a job step the DD name that is specified in the tracing DD parameter, all MAINVIEW SRM activity for the job step is traced or not traced (if you specify IGNOREDD).

TRACEDD is the same trace as produced by the TRACE parameter in the function definition in SMFUNCxx. IGNOREDD can be useful when a quick override to a MAINVIEW SRM rule is needed. The other three tracing DD parameters are primarily for use by BMC Software personnel in resolving problems.

The MAINVIEW SRM connections are not invoked into the operating system early enough for the following functions to be displayed in the DD name-activated trace:

DSNCHECK  
SMSACSDC  
SMSACSMC  
SMSACSSC  
SMSACSSG  
SMSACSTE

You can use the TRACE=YES parameter in the SMDIAGxx member to activate tracing for these or any other functions.

---

# Appendix A    **MAINVIEW SRM General Syntax Rules**

This appendix provides an overview of MAINVIEW SRM statement and parameter structures. The following topics are discussed:

Statements and Parameters .....	A-2
Parameter Relationships .....	A-4
Comparison Operators .....	A-6
Name Masking .....	A-6
Field-to-Field Comparisons .....	A-10
Selection Criteria Ordering.....	A-11

## Statements and Parameters

You code the various specifications that control MAINVIEW SRM in SET statements in the library members. A SET statement is composed of the keyword SET and one or more parameter specifications. A SET statement begins with the SET keyword and continues until the next SET keyword is found or the end of the member is reached.

The following rules apply:

- All text must be uppercase.
- The SET keyword must precede all parameters. There are no column restrictions. For example, the following statements are all valid:

```
column 1
|
SET POOL=TEST
      SET POOL=TEST
SET POOL = TEST
```

- Parameters are composed of keywords and associated values. There are two types of parameters: action and selection. Action parameters indicate how the MAINVIEW SRM function operates. Selection parameters indicate which resources (volumes, jobs, data sets, and so on) are affected by the MAINVIEW SRM function.
- Action parameters must be placed immediately after the SET operation keyword. For members SMPOOLxx, SMFLSTxx, and SMRLSTxx, all action parameters must be on the same line or a continued line. For other members (SMMSYSxx, SMCALSxx, SMVARSxx, and SMFUNCxx), action parameters can be listed on separate lines.

In all members the first line *must* be preceded with SET. In SMMSYS and SMEVNT, lines that are *not* preceded by SET, INC, or EXC use the default value SET (except continuations). In all other members, lines that are *not* preceded by SET, INC, or EXC use the default value INC (except continuations).

- Selection parameters must follow the action parameters and start on a new line. Selection parameters can be preceded by the keyword INC or EXC to indicate inclusion or exclusion (the default is INC).
- The parameter value is separated from the parameter keyword by an equal sign (=) and in some cases, a greater than (>) or less than (<) sign; spaces are optional.

- Most parameters have a single value specification: a keyword, a name or a name mask, or a number; for example:

```
MODE=ACT
EXC PGM=GL0125A
EXTENT>5
```

- All selection parameters and some action parameters can have a list of values, in which case the individual values are enclosed in parentheses and separated by commas or blanks:

```
POOL=( TEST , WORK , PROD )
```

- Parameter specifications within a single line are separated by commas or spaces; for example:

```
SET POOL=WORK , USEVOL=ALL
SET POOL=WORK USEVOL=ALL
```

- Variables are concatenated by placing them together, without the use of any special connective characters. Blanks or commas cannot be placed between concatenated variables; for example:

```
DSN=&TESTHLQ&DIVCODE&TESTLLQ
```

- Blank lines are allowed.
- In SMPOOL<sub>xx</sub> or a filter or rule list, a SET statement action parameter or selection parameter can be continued onto the next line by placing any character in column 72.
- A comment line is indicated by an asterisk in column 1.
- Comments can be embedded in a line by starting the comment with /\* and ending it with \*/. The following restrictions regarding /\* \*/ comments apply:

- They can only be used outside of a keyword and its associated value.
- They cannot be used within a parenthesized list of values.

Examples of valid /\* \*/ comments follow:

---

```
INC JOBNAME=TSTJOB1 DSN=TEST.DSN /* Comment is outside of any keyword and associated value */
INC JOBNAME=TSTJOB1 /* Comment is outside of any keyword and associated value */ DSN=TEST.DSN
```

---

Example of an invalid /\* \*/ comment follows:

```
INC   DSN= /* Comment within keyword and associated value not allowed */TEST.DSN
```

A /\* \*/ comment appearing within a parenthesized list of values is also invalid, as the following shows:

---

INC VOLSER = (DSN1.TESTA,	/* No comment allowed here within a list of values */	X
DSN1.TESTB,	/* No comment allowed here within a list of values */	X
DSN1.TESTC)		

---

If /\* \*/ comments are used in either of the two previously mentioned invalid forms, MAINVIEW SRM will not flag it as an error. Instead, MAINVIEW SRM will view the leading / in the /\* as a pattern matching character, causing it to always match any value, which is not what was intended.

## Parameter Relationships

Action parameters are independent of each other. They can be specified in any order without affecting their operation. For members SMFLST<sub>xx</sub> and SMRLST<sub>xx</sub>, all action parameters for a single SET statement must be on the same line (or on a continued line). For other members, action parameters can be on separate lines.

Selection parameters are cumulative; each selection parameter within a statement is modified by subsequent selection parameters. Selection parameters on a single line are related by an AND condition, whereas selection parameters on separate lines are related to preceding lines by an OR condition. For example:

```
VOL=TEST01 , DSN=APPROD . VNDR . MASTER
```

indicates that the volume ID must be TEST01 *and* the data set name must be APPROD.VNDR.MASTER.

```
VOL=TEST01  
DSN=APPROD . VNDR . MASTER
```

indicates that the MAINVIEW SRM function is applied if *either* the volume ID is TEST01 *or* the data set name is APPROD.VNDR.MASTER.



Multiple values for a single selection parameter are related by an OR condition; that is, when several values are specified for a single selection parameter by enclosing the list of values in parentheses, any of the specified values will satisfy the selection. The following examples demonstrate this relationship:

#### Example

This example indicates that the second name qualifier must be DBYTD *or* DBQTR *or* DBDLY:

```
DSN2 = (DBYTD, DBQTR, DBDLY)
```

When you specify multiple selection parameters with value lists, the selection is satisfied by a combination of OR and AND conditions, as follows:

```
DSN2 = (DBYTD, DBQTR)  DSN3 = (RECEIPT, ORDER)
```

The example indicates that the MAINVIEW SRM function is applied if the second name qualifier is either DBYTD *or* DBQTR *and* if the third name qualifier is either RECEIPT *or* ORDER.

Selection parameter specifications are usually contained within a single line; however, they can be continued onto multiple lines. Continuation is indicated by a non-blank character in column 72. Therefore, parameter specifications continued on the next line are treated as being connected by AND logic. For example, the following statement indicates that the data set name must be APPROD.VNDR.MASTER *and* the volume ID must be WORK for the MAINVIEW SRM function to be applied. If the second line was not a continuation line, the two parameters would be related by OR logic.

	column 72
DSN=APPROD.VNDR.MASTER	X
VOL=WORK	

## Comparison Operators

In selection parameters, the following are valid comparison operators. Operators for “less than or equal to” and “greater than or equal to” are not supported.

Operator	Description
= (equal)	value of the parameter at run time must equal the specified value
< (less than)	value of the parameter at run time must be less than the specified value
> (greater than)	value of the parameter at run time must be greater than the specified value
≠ (not equal to)	value of the parameter at run time must be a value other than the specified value (the logical NOT symbol preceding the equal symbol is EBCDIC X'5F')

## Name Masking

You can specify names or any character-field value in any MAINVIEW SRM parameter explicitly by using the complete name. You can also specify a partial name with special mask symbols to build flexible criteria. MAINVIEW SRM supports the following mask symbols:

- %** (percent) specifies a single numeric character; multiple mask characters can be specified and the mask can be embedded within text.
- VOL=PRODV% would be satisfied by PRODV1 or PRODV9 (but not PRODVX).
  - VOL=TST%V% would be satisfied by TST5V1 or TST2V9 (but not TSTAV5 or TST44V).
  - VOL=WRK%% would be satisfied by WRK104 or WRK582 (but not WRK33).
- ?** (question mark) specifies any character except a period (.), including numeric, alphabetic, or special characters (such as #); multiple mask characters can be specified, and the mask can be embedded within text.
- VOL=PR???? would be satisfied by PRODV1 or PR45B6 (but not PR64).

- VOL=PR??9 would be satisfied by PR4B9 or PR#39 (but not PROD4)

/ (slash) specifies zero or more characters of any type at the end of a name; this mask character terminates a name mask and any characters that follow are ignored.

**Note:** A period is significant in a DSN name mask. The parameter DSN=TEST/ differs from DSN=TEST/

- VOL=TST/ would be satisfied by TST, TSTA, TSTB14, or TST40V (but not TSXX44)
- VOL=TST%/ would be satisfied by TST4BX or TST477 (but not TSTA64)
- VOL=/ would be satisfied by any combination of characters
- DSN=TEST/ would be satisfied by TEST.CICS.PPT (but not PROD.CICS.PPT)
- DSN=TEST. / would be satisfied by TEST.CICS.PPT (but not TESTX.CICS.PPT)

\* (asterisk) specifies any character string in a single name node; this mask character can be embedded within other text.

**Note:** Leading asterisks can be confusing. Name matching stops after the name mask has been exhausted, even if more characters remain in the name. LLQ=\*T selects LAST but not TEST. The asterisk requires a non-zero number of characters. LAST has three characters (matching the asterisk) before the T (matching the T in the mask). TEST, however, has zero characters before a T is encountered; the initial T in TEST matches the T in the mask \*T, but because no characters precede the T, the match fails.

- DSN=AP1177A.\*.MASTER would be satisfied by AP1177A.VNDR.MASTER or AP1177A.VOUCHER.MASTER (but not AP1177A.VNDR.TEST.MASTER or AP1177A.MASTER)
- DSN=AP1177A.PER\*.MASTER would be satisfied by AP1177A.PER11.MASTER or AP1177A.PER07A.MASTER (but not AP1177A.MON12.MASTER)

- DSN=AP117A.\*TST.MASTER would be satisfied by AP117A.YEARTST.MASTER or AP117A.TST.MASTER (but not AP117A.DAILY.MASTER)

**\*\*** (double asterisk) specifies any non-zero number of name nodes; this mask character can be embedded within other text.

**Note:** Using **\*\*** at the end of a name string is equivalent to using **/**. For example, AP.TEST.MAS\*\* is equivalent to AP.TEST.MAS/. The **/** mask, however, cannot be used with following characters, as it terminates the mask. The value of the **\*\*** mask is that it can be followed by more name nodes.

- DSN=AP.\*.\*.MASTER would be satisfied by AP.VNDR.MASTER or AP.VNDR.TEST.MASTER or AP.VNDR.TEST.YTD.MASTER (but not AP.VNDR.MASTER.TEST or AP.MASTER)
- DSN=\*\*.\*.TEST.CICS would be satisfied by GLYTD5.TEST.CICS or GLYTD.PER04.TEST.CICS (but not TEST.CICS)
- DSN=AP.\*TST\*\*D.MASTER would be satisfied by AP.TST002.YTD.MASTER (but not AP.PR002.YTD.MASTER)

## Variables in Name Masks

The name of a variable can be used in any non-numeric MAINVIEW SRM selection parameter and in action parameters for the DSNCHECK function. The values of the variable are substituted into the parameter before the parameter is evaluated.

Variables are defined in the SMVARS.xx parmlib member. A variable can contain multiple values. For example, the variable APTEST is defined as follows:

```
SET VARIABLE = APTEST
VALUE = APTEST.*.PERIOD12
VALUE = APTEST.MASTER./
```

A selection parameter using the APTEST variable might be used as follows:

```
INC DSN = &APTEST
```

This include statement is equivalent to the following statements:

```
INC DSN = APTEST.*.PERIOD12
INC DSN = APTEST.MASTER./
```

You can concatenate variables. For example, the following variables are defined in SMVARSxx:

```
SET VARIABLE = APTESTHLQ
VALUE = APTEST
VALUE = APTST
VALUE = TESTAP*
SET VARIABLE = DIVISIONCODE
VALUE = RBK
VALUE = CRM
VALUE = DSR
SET VARIABLE = APLLQ
VALUE = TRAN/
VALUE = UPD/
VALUE = MASTER*
```

The variables from the previous example can be concatenated as follows in an FLST or RLST statement:

```
INC DSN = &APTESTHLQ&DIVISIONCODE&APLLQ
```

Do not use blanks or commas between concatenated variable names.

Observe the following rules when you use variables:

- Variables cannot be used with numeric parameters such as BLKSIZE or SECSPACE
- Variables can be used alone or concatenated with other variables. Variables *cannot* be mixed with text or name mask characters. For example:

INC DSN = &PRODHLQ	is valid
INC DSN = &PRODHLQ&ACCT	is valid
INC DSN = &PRODHLQ/	is invalid
INC DSN = &PRODHLQ.*.&PRODLLQ	is invalid
INC DSN = &PRODHLQ*.GLYTD.TRAN	is invalid

- Variables can contain any text and any mask characters. To use mask characters with variables, the mask characters must be defined within the variable. For example:

```
SET VARIABLE = PRODHLQ  
VALUE = ACCT*.*.*.  
SET VARIABLE = PRODLLQ  
VALUE = DAILY/  
VALUE = MONTHLY/  
INC DSN = &PRODHLQ&PRODLLQ
```

is equivalent to:

```
INC DSN = ACCT*.*.*.DAILY/  
INC DSN = ACCT*.*.*.MONTHLY/
```

- Variable names can be from 1 to 30 characters long, plus a preceding ampersand character (&). For example:

```
SET VARIABLE = FULLDATASETNAME  
VALUE = . . .  
INC DSN = &FULLDATASETNAME
```

- The ampersand character can only be used to precede a variable name in a selection or action parameter. It cannot be used as part of a text string because it flags the beginning of a variable name.
- Variables are concatenated by placing them together, without the use of any special connective characters. Blanks or commas cannot be placed between concatenated variables. For example:

```
DSN=&TESTHLQ&DIVCODE&TESTLLQ
```

- Variable definitions cannot be nested; that is, a variable definition that contains a variable is invalid.

## Field-to-Field Comparisons

Selection parameters can be compared to values in other selection parameters by preceding the target operand (on the right side of the equal sign) with an exclamation point. For example, comparisons enable you to select data sets where the high-level qualifier is equal to the user name or where the job name is equal to the user name. The following examples demonstrate this comparison.

INC HLQ = !DSN2	includes all data sets with a high-level qualifier that is equal to the second data set name node
INC HLQ = !USER	includes all data sets with a high-level qualifier that is equal to the user name
EXC JOB = !USER	excludes all data sets with a job name that is equal to the user name

The exclamation point is required to indicate the field-to-field comparison. If the exclamation point is omitted, the information to the right of the equal sign is interpreted as text. For example:

INC HLQ = !USER	compares the high-level qualifier to the user ID
INC HLQ = USER	compares the high-level qualifier to the literal USER

Field-to-field comparisons work for all selection parameters (INC and EXC statements) in all function FLSTs and RLSTs. Field-to-field comparisons do not work with action parameters (SET statements).

## Selection Criteria Ordering

To use MAINVIEW SRM successfully, you must fully understand the processing flow of selection parameters. Resources to be affected by MAINVIEW SRM functions are selected or excluded based on the selection parameters, including the order and interrelationships (OR, AND) of the selection parameters.

*Selection parameters are processed from top to bottom; as soon as a match to the selection parameters is made, the selection process stops. MAINVIEW SRM takes the first match to selection parameters. Therefore, selection parameters should be listed from specific to general. For example, the following specification selects all data sets with PROD1 as the first qualifier, except those that begin with PROD1.TEST:*

```
EXC DSN=PROD1 . TEST /
INC DSN=PROD1 . /
```

The separation onto two lines defines an OR relationship between the two criteria. A data set named PROD1.MASTER.FINAL is selected by the INC parameter because it does not match the criteria in the EXC parameter but it does match the INC parameter. However, a data set that is named PROD1.TEST.MASTER would not be selected; the EXC parameter would be satisfied first and the data set would be excluded.

If the sequence were reversed, as follows:

```
INC DSN=PROD1. /  
EXC DSN=PROD1. TEST /
```

PROD1.TEST.MASTER would not be excluded. The INC selection parameter is satisfied because PROD1.TST.MASTER begins with PROD1., and processing would halt before the EXC parameter was processed.

Definitions can vary and can be complicated. Remember these key points about selection parameters:

- those on the same line (or on continued lines) are related with an AND condition
- those on separate lines (without line continuation) are related with an OR condition.
- they should be listed from specific to general, because selection processing *stops* as soon as any INC condition is fully satisfied. A resource that has met the criteria in an INC statement is selected for processing and is not considered by any other filters or rules in the list. However, a resource that has met the criteria in an EXC statement is only excluded from the current SET statement and is passed to the next SET statement in the filter or rule list for consideration.

When you define resources to be selected, consider the following points. These principle apply to both filter and rule list entries.

- An EXC without an INC will not select any resources. For example:

```
SET SPACVOLA=50  
EXC DD=TRANSP /
```

excludes all data sets whose DD name begins with TRANSP. However, because there is no INC, no resources are selected for this rule; that is, there is no default INC following an EXC. To select all resources except those with TRANSP DD names, use an INC such as INC DD=/ after the EXC.

- A SET statement without an INC or EXC is ignored. For example:

```
SET VIO=YES  
SET VIO=NO  
INCLUDE SIZE<500KB
```



The first SET (VIO=YES) will never be executed, because no resources are included. All resources are passed to the second SET statement (VIO=NO) for consideration.



---

# Appendix B    Processing SMF Records

This appendix provides information about processing SMF records in MAINVIEW SRM. The following topics are discussed:

Overview .....	B-2
Using the PRSMF000 Utility .....	B-2
SMF Record Layout .....	B-3

## Overview

All MAINVIEW SRM message activity displays on the job log. In addition, these activity messages can be written to the SMF data set. The messages that are written to SMF contain the same information as those that are written to the job log and can be accumulated for any length of time. The messages can be

- extracted by the SMF ID code with the IBM IFASMFDP utility
- analyzed with the MAINVIEW SRM PRSMF000 utility
- reported on by function

**Note:** StopX37/II provides a separate logging facility that logs an SMF record for each attempted job recovery.

## Using the PRSMF000 Utility

The PRSMF000 utility sorts the messages by function and date, and it writes one PDS member per function. The MAINVIEW SRM ISPF interface enables you to view the processed SMF data.

**Note:** In Release 3.5 of ProSMS, the SMF record format was expanded to accommodate the StopX37/II recovery information that was used to generate the SMFX37B report. To provide compatibility between the old and new SMF record formats, program SMSMFCNV was added to the LOG ANALYSIS job flow. SMSMFCNV converts the new SMF record format (release 3.5 and later) to the old format (release 3.4 and earlier), which is processed by program PRSMF000 to produce the SMF message library. This program allows SMF message records from any release to be processed by PRSMF000. However, the ProSMS SMF message library should be viewed by using the ProSMS ISPF interface from the same release that created the records.

A sample of the required JCL for SMF record processing is supplied in the JCLSMFL member of the *?prefix.BBSAMP* library. The SMF record layout is also supplied. Figure B-1 “SMF Record Layout (Part 1 of 7)” on page B-3 shows an example of the record layout. The production of SMF messages is controlled by action parameters in the following members:

- |                 |   |
|-----------------|---|
| <b>SMMSYSxx</b> | The SMF record ID can be specified with the SMFID parameter. See SMFID in Appendix D, “Global Parameters.”      |
| <b>SMFUNCxx</b> | Function definitions control the level of produced messages. See the “Function Member Parameters” on page D-107 |

**SMFLSTxx** Message production can also be controlled at the level of resource selection. See “Using SMFLSTxx—Filter List Parameters” on page 5-2.

## SMF Record Layout

Figure B-1 shows the record layout for SMF.

**Note:** This macro is supplied as member @SMF on the product distribution tape data set *?prefix.BBSAMP*.

**Figure B-1** SMF Record Layout (Part 1 of 7)

---

```
@SMF DSECT
*
*   SMF REC - PROSMS SMF RECORD LAYOUT.
*
* DSECT NAME - SMFPRHDR (DEFAULT).
* CREATED BY - SMM90050.
*   SUBPOOL - N/A.
*   SIZE - SET BY CALLER OF SMFPRHDR.
* SERIALIZED - N/A.
* RESIDENCY - ENVIRONMENT DEPENDANT.
* POINTED TO - N/A.
*
*   FUNCTION - FORMAT OF THE RECORD TO BE WRITTEN TO SMF BY VARIOUS
*               PROSMS COMPONENTS.
*
SMFPRHDR DSECT
SMFPRLEN DC    H'0'          RECORD LENGTH FOR VB OR VBS.
SMFPRSEG DC    H'0'          SEGMENT DESCRIPTOR FOR VBS.
*
SMFPRFLG DC    B'10000010'   DEFAULT TO VS/2 XA.
SMFPRRRF EQU   X'80'          NEW SMF RECORD FORMAT.
SMFPRSUT EQU   X'40'          SUBTYPE FIELD IS USED.
SMFPRV4 EQU    X'10'          MVS/ESA V4.
SMFPRESA EQU   X'08'          MVS/ESA.
SMFPRVXA EQU   X'04'          MVS/XA.
SMFPROS EQU    X'02'          OS/VS2.
SMFPRBFY EQU   X'01'          BERRYMAN MACHINE (PR/SM).
*
SMFPRRTY DC    FL1'0'         RECORD TYPE (NUMBER).
SMFPRTME DC    FL4'00'        TIME IN HUNDRETHS REC WRITTEN.
SMFPRDTE DC    PL4'00'        DATE RECORD WAS WRITTEN.
SMFPRSID DC    CL4' '         SYSTEM ID FROM SMCASID IN SMCA.
*
*   END OF STANDARD SMF HEADER (W/O SUBTYPES)
```

**Figure B-1      Figure 13-1 SMF Record Layout (Part 2 of 7)**

```

*
SMFPRSSI DC      CL4 '      '      SUBSYSTEM ID.
*
SMFPRSTY DC      H'0'      RECORD SUBTYPE.
SMFSTMSG EQU      1      - MESSAGE SMF RECORD.
SMFSTREC EQU      2      - RECOVERY PERFORMED.
*
*      END OF STANDARD SMF HEADER (W/ SUBTYPES)

*
SMFPRTRN DC      AL2(0)      # OF TRIPLETS IN RECORD.
DC      H'0'      RESERVED.
*
*      "TRIPLET" IS A TERM FOR 3 CONTIGUOUS FIELDS THAT DESCRIBE A
*      SUB-SECTION OF THIS RECORD.  THE 3 FIELDS ARE THE OFFSET FROM THE
*      BEGINNING OF THE RECORD, THE LENGTH OF A DATA SECTION, AND THE
*      COUNT OF THE NUMBER OF OF DATA SECTIONS THAT MAKE UP THE
*      SUB-SECTION.  IT IS POSSIBLE THAT A SUB-SECTION WILL NOT BE
*      GENERATED.  IN THAT CASE THE COUNT WILL BE ZERO.  FOR ANY RECORD
*      WITH TRIPLETS, THE LAST PART OF THE HEADER SECTION BEGINS WITH
*      THE FIRST TRIPLET.  THIS TRIPLET DESCRIBES THE PRODUCT SECTION.
*
SMFPRTRP DS      0X      ** PRODUCT SECTION TRIPLET.
SMFPRPRS DC      AL4(0)      OFFSET TO THE PRODUCT SECTION.
SMFPRPRL DC      AL2(0)      LENGTH OF PRODUCT SECTION.
SMFPRPRN DC      H'1'      NUMBER OF SECTIONS (ALWAYS 1).
*
SMFPRJCS DS      0X      ** JOB INFORMATION CONTROL SECTION
SMFPRJIO DC      AL4(0)      OFFSET TO THE JOB INFO SECTION.
SMFPRJIL DC      AL2(0)      LENGTH OF JOB INFO SECTION.
SMFPRJRN DC      H'1'      NUMBER OF SECTIONS (ALWAYS 1).
*
SMFPRMSG DS      0X      ** MESSAGE CONTROL SECTION.
SMFPRMCS DC      AL4(0)      OFFSET TO THE MSG CNTL SECTION.
SMFPRMCL DC      AL2(0)      LENGTH OF MSG CNTL SECTION.
SMFPRMCN DC      Y(0)      NUMBER OF SECTIONS.
*
SMFPRDAS DS      0X      ** DATA SECTION.
SMFPRASS DC      AL4(0)      OFFSET TO THE DATA SECTION.
SMFPRASL DC      AL2(0)      LENGTH OF THE DATA SECTION.
SMFPRASN DC      Y(0)      NUMBER OF SECTIONS.
*
SMFPRDAC DS      0X      ** DATA CONTROL SECTION.
SMFPRDCS DC      AL4(0)      OFFSET TO THE DATA CNTL SECTION
SMFPRDCL DC      AL2(0)      LENGTH OF THE DATA CNTL SECTION
SMFPRDCN DC      Y(0)      NUMBER OF SECTIONS.
*
      ORG      ,      MAKE SURE LOC CNTR AT MAX.
      DS      0F      FILL TO FULLWORD.
*

```

**Figure B-1      Figure 13-1 SMF Record Layout (Part 3 of 7)**

SMFPRHDR_LEN	EQU (*-SMFPRHDR)	L(HEADER SECTION).
SMFPRTRP_LEN	EQU (*-SMFPRTRP)	L(TRIPLET SECTION).
SMFPRTR# EQU	(*-SMFPRTRP)/8	MAX COUT OF TRIPLETS.

\*  
\*      PROSMS PRODUCT CONTROL SECTION.  
\*

SMFPRPRO DSECT		
SMFPRPRD DC	CL8'PROSMS'	PRODUCT NAME.
SMFPROVS DC	X'00'	PRODUCT VERSION NUMBER.
SMFPRMPRE DC	XL3'00'	MESSAGE PREFIX
SMFPRMODE DC	XL1'00'	MODE
SMFPRRS1 DC	XL3'00'	RESERVED.
SMFPRMVS DC	CL8' '	MVS SOFTWARE LEVEL FROM CVT-40.
SMFPRXNM DC	CL8' '	SYSPLEX NAME (FROM ECVTSPLX).
SMFPRSNM DC	CL8' '	NAME OF THE CURRENT SYSTEM.
SMFPRFUN DC	CL8' '	FUNCTION NAME.
SMFPRMOD DC	CL8' '	MODULE NAME.
SMFPRCSN DC	CL8' '	CSECT NAME.
SMFOSYS DC	CL12' '	OS ID.
SMFOSFM DC	CL12' '	FMID.
SMFDFPL DC	CL12' '	DFP LEVEL.
SMFCPUM DC	CL12' '	CPU MODEL.
SMFCPUS DC	CL12' '	CPU SERIAL NUMBER.
SMFTSOE DC	CL12' '	TSO/E LEVEL.
SMFSSID DC	CL4' '	SUBSYSTEM ID USED BY LOADER.
SMFPRSCL EQU	(*-SMFPRPRO)	LENGTH OF PRODUCT SECTION.

\*  
\*      PROSMS JOB INFORMATION SECTION.  
\*

SMFPRJOB DSECT		
SMFJNAME DC	CL8' '	JOB NAME.
SMFPRJID DC	CL8' '	JES JOB ID.
SMFPRJS DC	CL8' '	CURRENT JOBSTEP.
SMFPRJPS DC	CL8' '	CURRENT JOB PROCSTEP.
SMFPRJSP DC	CL8' '	PROGRAM NAME ON EXEC CARD.
SMFRGRP DC	CL8' '	RACF GROUP
SMFRUID DC	CL8' '	RACF USERID
SMFJST DC	F'0'	JOB START TIME
SMFJSD DC	F'0'	JOB START DATE
SMFJCLAS DC	C' '	JOB CLASS
DS	0D	DOUBLEWORD ALIGNMENT.
SMFPRJSL EQU	(*-SMFPRJOB)	LENGTH OF PRODUCT SECTION.

\*  
\*      - - - - -  
\*      M E S S A G E   C O N T R O L   S E C T I O N   ( M C S )  
\*

PSMPMCS DSECT		
PSMMSGLN DC	Y(0)	LENGTH OF MESSAGE.

**Figure B-1      Figure 13-1 SMF Record Layout (Part 4 of 7)**

```

PSMMSGRI DC      Y(0)                      RESERVED.
PSMMSGTX DS      0C                      START OF MESSAGE TEXT.
*
* - - - - -
*           D A T A   C O N T R O L   S E C T I O N   ( D C S )
*
*   THIS IS BASICALLY A COPY OF THE GWA.  USE THE DCS TRIPLET TO
*   DETERMINE THE AVAILABLE SET OF FIELDS ALONG WITH A DETERMINATION
*   OF ITS INCLUDING INTO THE RECORD.  IF THE LENGTH IS ZERO, IT IS
*   NOT IN THE RECORD.
*
*   -- NOTE -- ANY CHANGE MADE TO THE GWA IN THE COPY BOOK SMM4WORK
PSMPDCS DSECT
PSMMSGLEN EQU 256                      MESSAGE BUFFER LEN - MUST BE
*                                       SAME AS PMSGLEN!!!!!!
*
PSMEYE DS CL4                          SMM4
PSMRDW DS H                            LENGTH OF RECORD
PSMRDWZ DS H                          HALFWORD INITIALIZED TO ZERO
PSMSMUV DS F                          SMUV ADDRESS
PSMSMAT DS F                          SMAT ADDRESS
PSMSMFNM DS C                          SMF "NAME" FIELD
PSMSMFRT DS C                          SMF RECORD TYPE
PSMSMFTI DS XL4                        TIME OF RECORD
PSMSMFDA DS XL4                        DATE OF RECORD
PSMSMFSI DS CL4                        SYSTEM IDENTIFICATION (SID)
PSMSMFJN DS CL8                        JOB NAME
PSMSMFET DS XL4                        ENTRY TIME TO RDR (.01 SECS)
PSMSMFED DS XL4                        ENTRY DATE TO RDR (00YYDDDF)
PSMX37 DS CL4                          C'X37'
PSMVERS DS CL2                        RELEASE/VERSION NUMBERS
PSMCPUTI DS F                          CUMMULATIVE CPU TIME
PSMSTEPT DS A                          STEP CPU TIME
PSMERROR DS C                          Y=RECOVERY NOT ATTEMPTED
PSMFFLAG DS C                          TYPE OF RECOVERY
PSMFRED EQU 00                          REDUCE (SPACSECR)
PSMFSEC EQU 04                          ADD SECONDARY (SPACSECA)
PSMFSWI EQU 08                          VOLUME ADD (SPACVOLA)
PSMFSNA EQU 12                          REDUCE PRIMARY (SPACPRIM)
PSMFRCAT EQU 16                          RECAT (NOCATLG2)
PSMFVRED EQU 20                          VSAM REDUCE (SPACSECR)
PSMFSECI EQU 24                          SECNDRY INCREASE (SPACSECI)
PSMFSECB EQU 28                          SECNDRY BESTFIT (SPACSECB)
PSMFSWIR EQU 32                          VOL ADD SECRED (SPACSWIR)
PSMSKIP DS C                            SKIP FLAG BITS
PSMSKIPEXCP EQU X'80' 1... .. SKIP EXCP CHECK
PSMSKIPNOTE EQU X'40' .1... .. SKIP NOTE/POINT CHECK
PSMSKIPDISP EQU X'20' ..1. .... SKIP DISP CHECK
PSMSKIPDSN EQU X'10' ...1 .... SKIP DSNAME CHECK
PSMSKIPENQU EQU X'08' .... 1... SKIP DSNAME ENQ CHECK

```



**Figure B-1      Figure 13-1 SMF Record Layout (Part 5 of 7)**

PSMSKIPDC	EQU	X'04'	.... .1..	SKIP DEVICE CHARACTERISTICS
PSMSKIPCTG	EQU	X'02'	.... ..1.	SKIP CONTIG CHECK
*SPARE	EQU	X'01'	.... ...1	SPARE
*				
PSMSTEPN	DS	AL1		STEP NUMBER
PSMDATE	DS	F		CURRENT DATE
PSMTIME	DS	F		CURRENT TIME
PSMBTIME	DS	F		CURRENT TIME (BINARY)
PSMYEAR	DS	F		CURRENT YEAR
PSMNDAY	DS	F		CURRENT # OF DAYS SINCE
PSMCDAY	DS	CL10		CURRENT DAY
PSMCTIME	DS	CL8		CURRENT TIME (CHARACTER)
PSMSTYP	DS	CL3		SPACE ALLOCATION TYPE
PSMIDCVS	DS	C		C'Y' = REDUCE PRIMARY FOR
*				
NON-JCL-ALLOC VSAM (IDCAMS)				
*				
* INFORMATION GATHERED FOR USE DURING SELECTION PROCESSING.				
*				
PSMJNAME	DS	CL8		NAME OF JOB
PSMJACT1	DS	CL20		JOB ACCT #1
PSMJACT2	DS	CL20		JOB ACCT #2
PSMJACT3	DS	CL20		JOB ACCT #3
PSMSACT1	DS	CL20		STEP ACCT #1
PSMSACT2	DS	CL20		STEP ACCT #2
PSMSACT3	DS	CL20		STEP ACCT #3
PSMPGMNM	DS	CL20		PROGRAMMER NAME FIELD
PSMRGRP	DS	CL8		RACF GROUP
PSMRUID	DS	CL8		RACF USERID
PSMACF2_USERID	DS	CL24		ACF2 USERID
PSMOWNER	DS	CL40		VSAM OWNER ID
PSMDSNM	DS	CL44		NAME OF DATASET
PSMCAT	DS	CL44		CATALOG NAME DATASET IS IN
PSMDSORG	DS	CL3		DATASET ORGANIZATION
PSMDSTYP	DS	CL4		DATASET TYPE (PERM/TEMP/GDG)
PSMUNITT	DS	CL8		UNIT TYPE (IE. 3350)
PSMDDNM	DS	CL8		ORIGINAL DDNAME
PSMJTYP	DS	CL3		TYPE OF TASK (JOB,STC,TSU)
PSMDISP1	DS	CL3		DATASET DISPOSITION
PSMLTYPE	DS	CL3		ALLOCATION TYPE (CYL,TRK,BLK)
PSMDISP2	DS	CL7		SCHED. DISP (CATLG,KEEP...)
PSMDISP3	DS	CL7		ABTERM DISP (CATLG,KEEP...)
PSMREL	DS	CL1		C'Y' - RLSE CODED IN SPACE PAR
PSMMDAY	DS	CL10		JOB START DAY OF WEEK
PSMMJDAT	DS	CL6		JOB START JULIAN DATE
PSMMDATE	DS	CL8		JOB START DATE (CHARACTER)
PSMMTIME	DS	CL8		JOB START TIME (CHARACTER)
PSMEXPDT	DS	CL8		EXPIRATION DATE
PSMRETPD	DS	F		RETENTION PERIOD
PSMPROG	DS	CL8		PROGRAM NAME ON EXEC CARD

**Figure B-1      Figure 13-1 SMF Record Layout (Part 6 of 7)**

PSMRECOV	DS	CL8	TYPE OF RECOVERY
PSMUNITN	DS	CL8	UNIT NAME
PSMOPER	DS	C	OPER SUPPLIED VOLUME FOR SWITC
PSMJCLAS	DS	C	JOB CLASS
PSMTMDSN	DS	C	TEMP DATASET FLAG (Y=YES)
PSMVOLSQ	DS	H	VOLUME SEQUENCE
PSMNVOL	DS	F	NUMBER OF VOLUMES
PSMPSPC	DS	F	PRIMARY SPACE ALLOCATION
PSMSSPC	DS	F	SECONDARY SPACE ALLOCATION
PSMJST	DS	F	JOB START TIME
PSMJSD	DS	F	JOB START DATE
PSMNUNIT	DS	F	NUMBER OF UNITS
PSMNQUAL	DS	F	NUMBER OF DSNAME QUALIFIERS
PSMHLQ	DS	0CL8	HLQ OF DSNAME
PSMQUAL1	DS	CL8	1ST QUALIFIER
PSMQUAL2	DS	CL8	2ND QUALIFIER
PSMQUAL3	DS	CL8	3RD QUALIFIER
PSMQUAL4	DS	CL8	4TH QUALIFIER
PSMQUAL5	DS	CL8	5TH QUALIFIER
PSMQUAL6	DS	CL8	6TH QUALIFIER
PSMQUAL7	DS	CL8	7TH QUALIFIER
PSMQUAL8	DS	CL8	8TH QUALIFIER
PSMQALL	DS	CL8	LAST QUALIFIER
PSMQUALS	EQU	*-PSMHLQ	LENGTH OF QUALS
PSMSTEP	DS	CL8	STEP NAME
PSMPROCS	DS	CL8	PROC STEP NAME
PSMCDCT	DS	XL60	DCT FOR CURRENT UCB
PSMMSGB	DS	CL(PSMMSGBLEN)	DEFAULT BUFFER FOR MSGED
*			
* Note: Actual message text begins at PSMMSGB+8. The MSGPREF			
* global is not included in the message ID (XXXnnnna).			
*			
PSMTEXT	DS	F	TOTAL NUMBER OF EXTENTS
PSMPOOL	DS	CL8	EASY/POOL POOL NAME FOR VOLUME
PSMREFV	DS	CL6	EASY/POOL REF VOLID
PSMLIB	DS	C	DSNTYPE=LIBRARY (PDSE)
PSMIAM	DS	C	IAM DATASET (Y/N)
PSMSIZE	DS	F	PRIMARY SPACE IN KB
PSMSIZES	DS	F	SECONDARY SPACE IN KB
PSMMSIZE	DS	F	1 PRI SPACE + 15 SECS IN KB
PSMCSPC	DS	F	CURRENT SPACE USED + NEW EXTEN
PSMLRECL	DS	F	DSN LRECL
PSMBLOCK	DS	F	DSN BLOCKSIZE
PSMRECFM	DS	CL3	RECORD FORMAT
PSMSTRIP	DS	C	STRIPED DATASET (Y/N)
PSMSTRTY	DS	CL2	STRIPED DATASET TYPE
PSMSTRCT	DS	F	STRIPED DATASET STRIPE COUNT
PSMHIER	DS	C	HIERARCHICAL DATASET (Y/N)
PSMROUND	DS	C	'Y' - ROUND CODED IN SPACE PAR
PSMSPEC	DS	C	'Y' - SPECIFIC ALLOCATION

**Figure B-1      Figure 13-1 SMF Record Layout (Part 7 of 7)**

```

PSMSMS13 DS      C                      'Y' - DFSMS 1.3 OR HIGHER
*
*   MISC FIELDS
*
PSMDVOL  DS      CL6                     VOLUME TO SWITCH TO
PSMRNVOL DS      H                      NO. OF ORIGINAL VOLS (RECAT)
PSMRVOL  DS      CL6                     ORIGINAL VOLUME NAME (RECAT)
PSMORGK  DS      F                      ORIGINAL PSW KEY ON ENTRY
PSMADDV  DS      CL1                     NEED TO ADD VOLUME TO JFCB
PSMAVGR  DS      CL1                     AVGREC KEYWORD IN EFFECT
PSMVCLST DS      CL44                    VSAM CLUSTER DSNAME
PSMVCOMP DS      CL5                     VSAM COMP TYPE (DATA OR INDEX)
PSMVTYPE DS      CL10                    TYPE OF VSAM DATA SET
PSMVDTYP DS      CL4                     TYPE OF VSAM CLUSTER
PSMDYNAL DS      C                      C'Y' - DYNAMIC ALLOCATION FLAG
PSMEFLAG DS      C                      C'Y'-MESSAGE IS ALREADY EDITED
*
*   Note:  Proprietary fields follow.  The length of the PSMPDCS
*           record is 2112 bytes.  (PFMSMF equates to x'840'.)
*
PSMSMF   EQU      (*-PSMPDCS),(*-PSMPDCS)  END OF SMF RECORD

```



---

## Appendix C    Altering Messages with the Message Exit

Some messages can be modified at execution time by using the MAINVIEW SRM message exit. The message exit can perform the following actions:

- suppress the message completely
- change the message text, including lengthening the message
- specify or alter the MCS flags
- specify or alter the descriptor codes
- specify or alter the routing codes

On entry to the message exit, registry 1 contains the address of a two-word parameter list:

- +0 is the address of the WTO work area (described below).
- +4 is the address of the MAINVIEW SRM SMUV (included as member @SMUV *Installation and Definition Reference* on the installation data set ?*prefix*.BBSAMP).

---

```
LEN          DC  Y(0)  LENGTH OF WTO WORK AREA.
MCSFLAGS     DC  XL2'00' MCS FLAGS.
TEXT         DC  0C   START OF TEXT.
```

\*

```
*THE FOLLOWING TWO FIELDS ARE NOT INCLUDED IN THE LENGTH INDICATED
*IN THE FIELD LEN. WHEN LENGTHENING THE MESSAGE, MAKE SURE YOU
*SAVE THE FOLLOWING TWO FIELDS. AFTER LENGTHENING, RESTORE THEM TO
*THE END OF THE MESSAGE TEXT.
```

\*

```
DESCODE      DC  XL2'00' DESCRIPTOR CODES.
ROUTCODE     DC  XL2'00' ROUTING CODES.
```

---

---

Note that this work area is not a fixed-length control block. Its length can be determined by using the LEN field. The maximum length of a full-length message, including all supporting fields, is 256 bytes. If the message is lengthened, the trailing routing and descriptor codes must be moved as well.

Changes to the routing code are ORed with ROUTCDE=15.

Your user-written exit must use standard OS/390 linkage conventions.  
Registers at entry are

R1	address of two-word parameter list
R13	address of an 18-word register save area
R14	return address back to MAINVIEW SRM message processor
R15	entry point address of SMMSGEXT

The MAINVIEW SRM message processor uses the return code in register 15 on return from SMMSGEXT to determine the disposition of the message:

R15 = 0	issue message and log to SMF
R15 < 8	do not issue message, but log to SMF
R15 > or = 8	do not issue message or log to SMF

SMMSGEXT must be reentrant and capable of executing in 31-bit mode.  
BMC Software Inc. recommends the following linkage editor instructions:

---

```
INCLUDE OBJECT(SMMSGEXT)
ENTRY SMMSGEXT
  MODE AMODE(31),RMODE(ANY)
  SETCODE AC(1)
  NAME SMMSGEXT(R)
```

---

A sample message exit is provided in member SMMSGEXT in the installation library *?prefix.BBSAMP*.

---

# Appendix D      Global Parameters

This appendix contains quick-reference lists of MAINVIEW SRM parameters. For an explanation of how to use global parameters, see Chapter 3, “Defining the MAINVIEW SRM System.”

SMMSYSxx - Master System Member Parameters . . . . .	D-2
SMPOOLxx - Pool Member Parameters . . . . .	D-99
SMSPOLxx - SMS Subpool Member Parameters . . . . .	D-103
SMCALsxx - Calendar Member Parameters . . . . .	D-104
SMVARsxx - Variable Member Parameters. . . . .	D-106
SMFUNCxx - Function Member Parameters . . . . .	D-107
SMDIAGxx - Diagnostic Member Parameters . . . . .	D-112
SMEVNTxx - Event Member Parameters . . . . .	D-113
SMVSCFxx - VTOC Scan Facility Parameters . . . . .	D-116

# Master System Member Parameters

**SMMSYSxx** SMMSYSxx contains the master system parameters for the MAINVIEW SRM system. Default parameter values can be specified that apply to the overall operating environment. For an explanation of the master system member, see Chapter 3, “Defining the MAINVIEW SRM System.”

## Subordinate Members

SMPOOLxx, SMSPOLxx, SMCALSxx, SMVARSxx, SMDIAGxx,  
SMFUNCxx

## Other Parmlib Members

SMEVNTxx (Automation)  
SMCRITxx (Critical List of Volumes)

## Parameter Quick Reference

Table D-1 provides a brief description of SET statement master system parameters. Detailed descriptions of each parameter are listed in alphabetical order after the table. The page is cross-referenced in the page number column.

**Note:** Unless otherwise defined, K, M, G, and T (kilobytes, megabytes, gigabytes, and terabytes) can be specified optionally along with a value in numeric parmliib member fields. After the value is converted to bytes, it is checked against system-defined minimum and maximum settings.



Table D-1 SET Statement System Parameters (Part 1 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
AOO_SUBSYS=xxxx or AOO_SUBSYS=(xxxx,xxxx,xxxx)	N	X							D-23	specifies the AutoOPERATOR subsystems that are to receive events
AUTO_MXTSK=nn	N					X			D-23	specifies the maximum number of tasks to be used in collecting volume space or data set information in response to automation requests for any AUTO function
AUTOJ_OINDX=xxxxxxxx	N					X			D-23	specifies the prefix value for the data set name that is appended to the AUTOJCL.DYHMMDD.T HMMSSTT to create the skeleton JCL data set name
AUTOPROC=xxxxxxxx	N							X	D-24	specifies the name of the cataloged procedure used to start SG-Auto
BBI3_SSID=xxxx	Y	X							D-25	specifies the CAS subsystem name to which the SVOS PAS should connect
BCDSn=xxxxxxxxxxx	N						X		D-25	specifies HSM CDS database files to be used by MAINVIEW SRM
BLKINPUT=Y/N	N		X						D-25	changes block size for input data sets
BLKOLDSR=Y/N	N		X						D-26	changes blocksize for output data sets opened with disposition of old or shared
CAL=xx	Y		X				X		D-26	suffix of parameter member SMCALSxx
CHECK=FIRST/ALLVOLS	N		X						D-26	specifies whether to check all volumes that the job requests during allocation or only the first volume that is requested

Table D-1 SET Statement System Parameters (Part 2 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
CNFG_MXTSK= <i>nn</i>	N	X							D-26	indicates the maximum number of tasks that the configuration component is to use to collect configuration data
CRITLIST= <i>xx</i>	N		X	X					D-27	specifies the suffix of an SMCRTxx parameter member
DADSMEX= <i>Y/N</i>	N						X		D-27	determines if the DADSM preprocessing exit (IGGPREF00) is called
DASDGENR=( <i>xxxxxxxx,...</i> )	N		X						D-28	specifies DASD generic names (1–8 characters) to be processed
DATEFMT= <i>MMDD/DDMM</i>	N			X					D-28	date format
DCTYPE=( <i>xxxxx,...</i> )	N				X				D-29	allows choice of one or more device characteristics to be maintained during volume switching (CACHE, SHARED, DUALCOPY, FASTWRITE)
DFREORGPRC= <i>xxxxxxxx</i>	N				X				D-29	defines default SPACVOLA reorganize-procedure name
DIAG= <i>nn</i>	N	X							D-29	suffix of parameter member SMDIAGxx
DIAGMSDD= <i>xxxxxxxx</i>	N	X							D-30	established WTO message tracing
DISPLAY= <i>ALL/LIC</i>	N	X							D-30	display functions list in the ISPF interface
DMYUNIT=( <i>xxxxxxxx, zzzzzzzz,....</i> )	N		X						D-30	unit name conversion (1–8 characters for each unit)
DP_RENAME= <i>Y/N/A</i>	N		X						D-31	determines if DASDPOOL is processed for volume selection during DADSM RENAME

Table D-1 SET Statement System Parameters (Part 3 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
DUMPDD=xxxxxxxx	N	X							D-31	provides SYS1.DUMPxx dump if MAINVIEW SRM abends
EVNT=xx	Y	X							D-32	specifies the suffix you assign to the name of the SMEVNTxx event definition member
FDRIAM=Y/N	N				X				D-32	for IAM customers only, FDRIAM=Y determines whether a data set is an IAM data set
FUNC=xx	Y	X							D-33	suffix of parameter member SMFUNCxx
HISTDAYS=nn	N						X		D-34	number of days (0-14) specified for gathering historical performance data
HLOGAUTH=nn	Y						X		D-34	automatic DFHSM log switch interval in hours
HLOGAUTM=nn	N						X		D-35	automatic DFHSM log switch interval in minutes
HLOGCOLL=Y/N	N						X		D-35	activation of DFHSM logfile data collection
HLOGINDX=xxxxxxxx	N						X		D-35	DSN prefix of DFHSM log extract file
HLOGLIM=xxxx	N						X		D-35	limits the number of log extract data sets that are allocated for processing one report request
HLOGPRIM=nnnn	N						X		D-36	size of primary allocation of log extract file
HLOGTASK=xxxxxxxx	N						X		D-37	name of proc to run on DFHSM logfile swap
HLOGUNIT=xxxxxxxx	N						X		D-37	unit name for allocation of log extract file
HLOGYDSN=xxxxxxxx	N						X		D-37	data set name of DFHSM logfile Y
HSMACTID=xxxxxxxx	N						X		D-38	high-level name of DFHSM activity data sets

Table D-1 SET Statement System Parameters (Part 4 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
IGNOREDD=xxxxxx	N	X							D-38	suppresses MAINVIEW SRM activity for job step
JCLEXT=Y/N	N		X						D-39	provides volume and unit information after accessing the catalog
JCLUREQ=Y/N	N		X						D-39	determines whether the UNIT information is required in the JCL
MAXVOL=nn	N				X				D-40	limits the number of volumes a data set is allowed to use
MCDSn=xxxxxxxxxx	N						X		D-41	specifies HSM migrated data set file that is allocated at system startup
MODTRCDD=xxxxxxxx	N	X							D-41	sets module entry/exit tracing
MREDUCE=Y/N	N				X				D-41	determines if secondary space reduction can occur on multivolume data sets allocated by JCL
MSGID=Y/N	N	X							D-42	specifies the inclusion of the MAINVIEW SRM message identifier in the message text
MSGLVL=I/W/E/S	N	X							D-42	specifies the level of messages to be generated
MSGPREF=xxx/SVM	N		X	X	X		X		D-42	MAINVIEW SRM message identifier prefix
NOCATDYN=Y/N	N				X				D-43	allows NOCATLG2 to process dynamically allocated data sets
NOCATPFX=xxx	N				X				D-43	second-level qualifier to be used when renaming a data set during NOCATLG2 processing

Table D-1 SET Statement System Parameters (Part 5 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
NOCATPRG=Y/N	N				X				D-43	allows data sets to be scratched before the expiration date during NOCATLG2 processing
NOCATSEC=xxxxxx	N				X				D-44	level of security performed before scratching or renaming a data set during NOCATLG2 processing (NONE, CREATE, READ, UPDATE, ALTER)
NOCATSMS=Y/N	N				X				D-44	allows SMS-managed data sets to be renamed, uncataloged, or scratched during NOCATLG2 processing
NOCATTIM=nn	N				X				D-45	specifies the amount of time (in minutes) that StopX37/II will wait while attempting to perform NOCATLG2 processing (DELETE option only) on an SMS-managed data set
NOCATVOL=SAME/DIFF	N				X				D-45	allows a new data set to be allocated to the same volume it was previously cataloged on during NOCATLG2 processing
NOCATWHEN=ALLOC/TERM	N				X				D-46	specifies when NOCATLG2 processing is to occur for non-SMS data sets (allocation or step termination)
OCDS=xxxxxxxxxxx	N						X		D-46	specifies HSM OCDS data set to be defined and allocated at system startup
OPMHLQ=xxxxxxxxx	N						X		D-46	high-level qualifier for MAINVIEW SRM output data set

Table D-1 SET Statement System Parameters (Part 6 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
ORIGDATA= <i>PRO/POOL</i>	N		X						D-47	specifies whether VOL and UNIT contain the original volser and unit values from the JCL or contain the current value
PASSWORD= <i>xxxxxxxxxx</i>	Y	X							D-47	specifies a MAINVIEW SRM password
PERFRM_PRC== <i>xxxxxxxxxx</i>	N						X		D-48	specifies the name of the procedure that is used to start the historical performance data collector
POOL= <i>xx</i>	Y	X							D-48	suffix of parameter member SMPPOOLxx
PROCOLD= <i>Y/N</i>	N		X						D-49	allows interception of DD statements that specify OLD allocations
REJECT= <i>FIRST/LAST</i>	N		X						D-49	controls termination of processing rejected data sets
REQTYPE= <i>Y/N</i>	N				X				D-49	specifies whether the MNTYPE statement is considered the request type instead of the mount type
RLS= <i>Y/N</i>	N						X		D-50	specifies whether the system should determine the VSAM record level sharing environment
SCAT= <i>STEPEND/IMMEDIATE</i>	N				X				D-50	forces immediate catalog update during volume switch
SG_INITPOOL= <i>nnnnnn</i>	N						X		D-50	specifies the maximum number of defined pools included in a single snapshot
SG_INITVOL= <i>nnnnnn</i>	N						X		D-51	specifies the maximum number of defined volumes included in a single snapshot

Table D-1 SET Statement System Parameters (Part 7 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
SG_IXFPNTVL= <i>nn</i>	N						X		D-51	specifies the number of hours between refreshes of the IXFP data tables
SG_MAXACCT= <i>nnnnn</i>	N						X		D-51	specifies the maximum number of active accounts in the application database
SG_MAXPOOL= <i>n</i>	N						X		D-51	specifies the number of pools that can be assigned to a volume
SG_MAXSSDSZ= <i>nnnnn</i>	N						X		D-52	specifies the maximum number of cylinders used for a solid-state disk drive
SG_READNTVL= <i>nnnn</i>	N						X		D-52	specifies the frequency (in minutes) at which the space collector scans the DASD volumes for historical space information to create a snapshot in memory
SG_RETRYLIM= <i>nnnn</i>	N						X		D-52	specifies the number of abend conditions that the space collector should ignore
SG_SIBSTK= <i>nn</i>	N						X		D-53	specifies the IXFP SIBBATCH parameter member to be used by the MAINVIEW SRM IXFP services for communications with the IXFP address space
SG_SPACHLDR= <i>mask</i>	N						X		D-53	defines a data set name mask that the space collector can use to identify space-holder data sets
SG_SUBTASKS= <i>nn</i>	N						X		D-53	specifies the number of volumes that can be read in parallel

Table D-1 SET Statement System Parameters (Part 8 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
SG_VVDSINFO= YES/NO	N						X		D-54	indicates whether the VVDS size and percentage used should be calculated for each volume that the space collector processes
SG_WRITNTVL=nnnn	N						X		D-54	specifies the frequency (in minutes) at which snapshots are written to the space database
SGA_ENQSCOP=GLOBAL/ LOCAL	Y							X	D-54	specifies the operational environment in which SG-Auto is to run
SGACMD=nn	N							X	D-54	specifies the two-position suffix of the initial command for executing the SG-Auto started task
SGASCAN= Y/N	N							X	D-55	specifies whether SG-Auto should be started in SCAN mode
SGASIM= Y/N	N							X	D-55	specifies whether SG-Auto should be started in SIMULATION mode
SGC_ADDEXIT=xxxxxxx	N						X		D-55	specifies the name of the Application Add Exit
SGC_CHKEXIT=xxxxxxx	N						X		D-55	specifies the name of the Application Check Exit
SGC_DEFEXIT=xxxxxxx	N						X		D-56	specifies the name of the Application Default Exit
SGC_KEYEXIT=xxxxxxx	N						X		D-56	specifies the name of the Application Account Code Build Exit
SGC_SECEXIT=xxxxxxx	N						X		D-56	specifies the name of the Application Security Exit
SGC_SELEXIT=xxxxxxx	N						X		D-56	specifies the name of the Application Select Exit



Table D-1 SET Statement System Parameters (Part 9 of 20)

Applies to										
Parameter			Allocation						Page	Description
	Required	All	EasyPOOL	EasySMS	StopX37/II	Automation	Reporting	SG-Auto		
SGC_STOGRP=Y/N	N						X		D-57	specifies whether to retrieve SMS storage-group information; this parameter should be set to YES only if SMS storage-group information is required in FLST or RLST processing
SGC_STORCLS=Y/N	N						X		D-57	specifies whether to retrieve storage-class information; this parameter should be set to YES only if SMS storage-class information is required in FLST or RLST processing
SGCDSN=xxxxxx...xxxxxx	Y						X		D-57	specifies the data set name for the dynamic allocation/deallocation of application database DD (SGCDB)
SGD_PROCNM=SGDCOLLS	N						X		D-57	specifies the name of the data-collector started task (1-8 characters)
SGD_SMFID=nnn	N						X		D-58	specifies an SMF record number for MAINVIEW SRM audit records that are written to the SMF data set for the space collector
SGDCOLLECT=Y/N	N						X		D-58	indicates the default for space pool collection; a corresponding parameter at the POOL level can be used to override the default
SGDCOLLECT $n$ =Y/N	N						X		D-58	specifies whether a pool is processed by the space alternate data collector that is assigned a suffix of $n$ , where $n$ is a number in the range of 1-8

Table D-1 SET Statement System Parameters (Part 10 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
SGDPROCNM <i>n=xxxxxxx</i>	N						X		D-58	specifies the cataloged procedure to be started for a specified copy of the space collector
SGDSMFID <i>n=</i>	N						X		D-59	specifies an SMF record number for MAINVIEW SRM audit records that are written to the SMF data set for a specified copy of the space collector
SGINITPOOL <i>n=</i>	N						X		D-59	specifies the maximum number of defined volumes to be included in a single snapshot for a specified copy of the space collector
SGINITVOL <i>n=</i>	N						X		D-60	specifies the number of hours between refreshes of the IXFP data tables for a specified copy of the space collector
SGMAXACCT <i>n=</i>	N						X		D-60	specifies the maximum number of active accounts in the application database for a specified copy of the space collector
SGMAXPOOL <i>n=n</i>	N						X		D-60	specifies the number of pools that can be assigned to a volume for a specified copy of the space collector
SGMAXSSDSZ <i>n=nnnnn</i>	N						X		D-61	specifies the maximum number of cylinders used for a solid state disk drive for a specified copy of the space collector
SGP_DSNINIT= <i>nnnnnnnn</i>	N						X		D-61	specifies the size, in number of data sets for the data set name index data space

Table D-1 SET Statement System Parameters (Part 11 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
SGP_EXITBBS= <i>nn</i>	N						X		D-61	specifies the number of megabytes to allocate in a scope common data space for the performance exit buffer block
SGP_EXITLIB=xxxxxx	N						X		D-62	specifies the default library where the performance collector SMF exits reside
SGP_MAXCCUS= <i>nnnn</i>	N						X		D-62	identifies the maximum number of control units that are in use during an interval
SGP_MAXDIRS= <i>nnnn</i>	N						X		D-62	identifies the maximum number of directors that are in use during an interval
SGP_MAXDSNS= <i>nnnn</i>	N						X		D-63	identifies the maximum number of data set names that are in use during an interval
SGP_MAXJOBS= <i>nnnn</i>	N						X		D-63	identifies the maximum number of jobs (batch, TSO, and started tasks) that are in use during an interval
SGP_MAXLCUS= <i>nnnn</i>	N						X		D-63	defines the maximum number of logical control unit/CHIP combinations in used during an interval
SGP_MAXPOLs= <i>nnnn</i>	N						X		D-63	identifies the maximum number of pools that are in use during an interval
SGP_MAXPTHS= <i>nnnn</i>	N						X		D-64	identifies the maximum number of CHPIDs that are in use during an interval
SGP_MAXPVLS= <i>nnnn</i>	N						X		D-64	identifies the maximum number of physical volumes that are in use during an interval

Table D-1 SET Statement System Parameters (Part 12 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
SGP_MAXRRKS=nnnn	N						X		D-64	identifies the maximum number of RAID ranks that are in use during an interval
SGP_MAXRSFS=nnnn	N						X		D-64	identifies the maximum number of RVA frames that are in use during an interval
SGP_MAXSCLS=nnnn	N						X		D-65	identifies the maximum number of storage classes that are in use during an interval
SGP_MAXVOLS=nnnn	N						X		D-65	identifies the total number of online DASD volumes on the OS/390 image being monitored
SGP_RDFCOMP=Y/N	N						X		D-65	specifies whether data compression is in effect for records being written to the performance resource data files
SGP_SIBSTK=xxxxxxxx	N						X		D-66	identifies the IXFP SIBBATCH parameter member to be used by the MAINVIEW SRM IXFP services for communications with the IXFP address space
SGP_SMF42=Y/N	N						X		D-66	specifies whether the SMF 42 record is to be written to the SMF data set
SGP_TRACE=xxxxxxx	N						X		D-66	specifies the trace default for the performance collector The default is NOTRACE
SGREADNTVLn=nnnn	N						X		D-67	specifies the frequency at which the space collector creates a snapshot in core for a specified copy of space database

Table D-1 SET Statement System Parameters (Part 13 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
SGRETRYLIMn=nnnn	N						X		D-67	specifies the number of abend conditions that the space collector should ignore for a specified copy of the space database
SGSPACHLDRn=xxxxxxxxxx	N						X		D-67	defines a data set name mask that the space collector can use to identify space holder data sets for a specified copy of the space database
SGSUBTASKSn=nn	N						X		D-68	specifies the number of volumes that can be read in parallel for a specified copy of the space database
SGWRITNTVLn=nnnn	N						X		D-68	defines the frequency at which snapshots are written to the space collector for a specified copy of the space database
SIZEISPRIM=Y/N	N		X		X				D-68	determines if the SIZE filter/rule list parameter includes only the size of the primary extent or the size of the primary and one secondary extent
SKIP=(CHECK=(xxx,xxx,...),DD name=xxxxxxxx, PROG=xxxxxxxx)	N				X				D-69	specifies checks to be bypassed during volume switching
SMFID=nnn	N		X	X	X		X		D-70	record number for MAINVIEW SRM SMF records
SMS_ALLOC=Y/N	N		X	X					D-70	determines if SMSSELCT is processed for SMSPOOL during DADSM ALLOCATE
SMS_EXTEND= Y/N	N		X	X					D-71	determines if SMSSELCT is processed for SMSPOOL_EXT during DADSM EXTENDNV

**Table D-1 SET Statement System Parameters (Part 14 of 20)**

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
SMSPOOL=xx	N		X	X					D-71	specifies the suffix of an SMSPOOLxx parameter member
START_ALL=Y/N	N	X							D-72	specifies whether all components should be started
START_ALLOC=Y/N	N		X	X	X				D-72	specifies whether the Allocation component should be started
START_AUTO=Y/N	N					X			D-72	specifies whether the automation component should be started
START_EHSM=Y/N	N						X		D-72	specifies whether the HSM collector subcomponent should be started
START_EPOOL=Y/N	N		X						D-72	specifies whether the EasyPOOL subcomponent should be started
START_ESMS=Y/N	N			X					D-73	specifies whether the EasySMS subcomponent should be started
START_RPRT=Y/N	N						X		D-72	specifies whether the Reporting component should be started
START_SGA=Y/N	N							X	D-73	specifies whether the SG-Auto subcomponent should be started
START_SGC=Y/N	N						X		D-73	specifies whether the application collector should be started
START_SGD=Y/N	N						X		D-74	specifies whether the space collector should be started
START_SGP=Y/N	N						X		D-74	specifies whether the performance collector should be started
START_X37=Y/N	N				X				D-74	specifies whether the StopX37/II subcomponent should be started

Table D-1 SET Statement System Parameters (Part 15 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
STKSCR=(xxx,xxx,xxx,xxx)	N		X						D-75	STK silo support
STOPX37II= YES/NO	N				X				D-75	specifies whether a full function or limited function of StopX37/II will be started at system start-up
STOPXONLY= YES/NO	N				X				D-76	specifies whether the StopX37/II will run without the other subcomponents of Allocation
SYSLIB=xxxxxxxxxxxxxx	N	X							D-76	specifies a default data set to be allocated at SVOS startup
SYSLIB2=xxxxxxxxxxxxxx	N	X							D-76	specifies a default data set to be allocated at SVOS startup
SYSLIB3=xxxxxxxxxxxxxx	N	X							D-76	specifies a default data set to be allocated at SVOS startup
TAPE_CA1DSN=xxxxxxxxxxxx	N						X		D-77	specifies the data set name of the CA1 data set
TAPE_CAT=(xxxxxxxx,xxxxxxxx,...)	N						X		D-77	specifies the tape management systems available for report generation
TAPE_CCTLTH=xxxxxxxxxx	N						X		D-77	specifies the high-level qualifier for the CONTROL-T data sets
TAPE_CHLQ=xxxxxxx	N						X		D-78	specifies the high-level qualifier for the TSCAN data sets
TAPE_COLHSM	N						X		D-78	specifies whether to collect DFHSM data
TAPE_CPRI=nnnn	N						X		D-78	specifies the number of cylinders for the primary allocation
TAPE_CSEC=nnnn	N						X		D-78	specifies the number of cylinders for the secondary allocation

Table D-1 SET Statement System Parameters (Part 16 of 20)

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
TAPE_CTLTRL=X	N						X		D-18	specifies the release number for Control T
TAPE_CVOL=(xxxxxx,xxxxxx,...)	N						X		D-79	specifies the volume serial numbers of the volumes to be used for the linear data sets, with a maximum of six volsers
TAPE_RMMDSN=xxxxxxxxxxx	N						X		D-79	specifies the data set name for the RMM control data set
TAPE_V_SUFF=xxxxxxxxxxx...	N						X		D-80	specifies one or more suffixes of the TAPECAT data set(s) used by the TSCAN image
TAPEGENR=(xxxxxxxxxx,...)	N						X		D-80	specifies tape generic names (1–8 characters) to be processed
TRACEDD=xxxxxxx	N	X							D-81	traces MAINVIEW SRM activity for jobstep
TRKCYL=nnnnn	Y		X		X				D-81	default device tracks per cylinder
TRKLEN=nnnnnnn	Y		X		X				D-82	default device bytes per track
USECAT=Y/N	N			X					D-82	ACS selection criteria catalog name usage
USEMVI=Y/N	N				X				D-82	specifies whether BBI3_SSID must be specified
VAR=xx	N	X							D-83	suffix of parameter member SMVARSxx
VSAM_ADJ4GB=YES/NO	N				X				D-83	controls StopX37/II recovery for non-extended VSAM data sets exceeding 4GB.
VSAMJCL=CLUS/COMP	N				X				D-83	controls level of processing of VSAM data sets



Table D-1 SET Statement System Parameters (Part 17 of 20)

Applies to										
Parameter			Allocation						Page	Description
	Required	All	EasyPOOL	EasySMS	StopX37/II	Automation	Reporting	SG-Auto		
VSAMLIMWARN=xx	N				X				D-84	specifies the percentage value to be used before issuing the 4 GB-limit message
VSAMPRIM= Y/N	N				X				D-84	use primary size for VSAM volume extensions
VSAMZSEC= Y/N	N				X				D-84	controls out-of-space recoveries for VSAM files with zero secondary space coded
VSCAN_AGER1=nnn	N				X				D-85	specifies the high end of age range one
VSCAN_AGER2=nnn	N				X				D-85	specifies the high end of age range two
VSCAN_AGER3=nnn	N				X				D-85	specifies the high end of age range three
VSCAN_AGER4=nnn	N				X				D-86	specifies the high end of age range four
VSCAN_AGER5=nnn	N				X				D-86	specifies the high end of age range five
VSCAN_AGER6=nnn	N						X		D-86	specifies the high end of age range six
VSCAN_AGER7=nnn	N						X		D-87	specifies the high end of age range seven
VSCAN_AGER8=nnn	N						X		D-87	specifies the high end of age range eight.
VSCAN_AGER9=nnn	N						X		D-87	specifies the high end of age range nine
VSCAN_PCTR1=nnn	N						X		D-88	specifies the high end of percentage used range one
VSCAN_PCTR2=nnn	N						X		D-88	specifies the high end of percentage used range two
VSCAN_PCTR3=nnn	N						X		D-88	specifies the high end of percentage used range three

**Table D-1 SET Statement System Parameters (Part 18 of 20)**

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
VSCAN_PCTR4= <i>nnn</i>	N						X		D-89	specifies the high end of percentage used range four
VSCAN_PCTR5= <i>nnn</i>	N						X		D-89	specifies the high end of percentage used range five
VSCAN_PCTR6= <i>nnn</i>	N						X		D-89	specifies the high end of percentage used range six
VSCAN_PCTR7= <i>nnn</i>	N						X		D-90	specifies the high end of percentage used range seven
VSCAN_PCTR8= <i>nnn</i>	N						X		D-90	specifies the high end of percentage used range eight
VSCAN_PCTR9= <i>nnn</i>	N						X		D-90	specifies the high end of percentage used range nine
VSCAN_SIZR1= <i>nnnnnnnn</i>	N						X		D-91	specifies the high end of size range one
VSCAN_SIZR2= <i>nnnnnnnn</i>	N						X		D-91	specifies the high end of size range two
VSCAN_SIZR3= <i>nnnnnnnn</i>	N						X		D-91	specifies the high end of size range three
VSCAN_SIZR4= <i>nnnnnnnn</i>	N						X		D-92	specifies the high end of size range four
VSCAN_SIZR5= <i>nnnnnnnn</i>	N						X		D-92	specifies the high end of size range five
VSCAN_SIZR6= <i>nnnnnnnn</i>	N						X		D-92	specifies the high end of size range six
VSCAN_SIZR7= <i>nnnnnnnn</i>	N						X		D-93	specifies the high end of size range seven
VSCAN_SIZR8= <i>nnnnnnnn</i>	N						X		D-93	specifies the high end of size range eight
VSCAN_SIZR9= <i>nnnnnnnn</i>	N						X		D-93	specifies the high end of size range nine

**Table D-1 SET Statement System Parameters (Part 19 of 20)**

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
VSCAN_MNTSK=nn	N						X		D-94	specifies the minimum number of tasks (TCBs) used by the VTOC scan to perform the collection
VSCAN_MXTSK=nn	N						X		D-94	specifies the maximum number of tasks (TCBs) used by the VTOC scan to perform the collection
VSCAN_OINDX=xxxxxxxxxx	Y						X		D-94	specifies the prefix name of the VTOC scan collection data set
VSCAN_OPRI=nnnn	N						X		D-95	specifies the primary allocation size in cylinders for the VTOC scan collection data set
VSCAN_OSEC=nnnn	N						X		D-95	specifies the secondary allocation size in cylinders for the VTOC scan collection data set
VSCAN_OUNIT=xxxxxxx	N						X		D-95	specifies the direct access device type of the VTOC scan collection data set
VSCAN_OVOL=xxxxxx	N						X		D-95	specifies the volume serial number of the VTOC scan collection data set
VSCAN_TPRI=nnnn	N						X		D-96	specifies the primary allocation size in cylinders for the VTOC scan temporary data set
VSCAN_TSEC=nnnn	N						X		D-96	specifies the set secondary allocation size in cylinders for the VTOC scan temporary data
VSCAN_TUNIT=xxxxxxx	N						X		D-96	specifies the device type for the VTOC scan temporary data set
VSCAN_TVOL=xxxxxx							X		D-97	specifies the volume serial number for the VTOC scan temporary data set

**Table D-1 SET Statement System Parameters (Part 20 of 20)**

Applies to										
Parameter	Required	All	Allocation			Automation	Reporting	SG-Auto	Page	Description
			EasyPOOL	EasySMS	StopX37/II					
WTODC= <i>n</i>	N	X							D-97	specifies the message descriptor codes
WTORC= <i>nn</i>	N	X							D-97	specifies routing codes assigned to message text (1-16)
X37POOL= <i>NEW/ORIG</i>	N				X				D-97	specifies which volume is used by StopX37/II to determine the pool name in EOVS processing
X37RLS= <i>YES/NO</i>	N				X				D-98	enables VSAM RLS data sets

## Usage Notes

SMMSYS $_{xx}$  is the global or master parameter member. It is read by MAINVIEW SRM during subsystem startup. The suffix specifications in SMMSYS $_{xx}$  indicate which versions of parameter members (SMPOOL $_{xx}$ , SMCALS $_{xx}$ , SMVARS $_{xx}$ , SMFUNC $_{xx}$ ) contain specifications for the current execution of MAINVIEW SRM.

SMMSYS $_{xx}$  parameters can be changed by editing the member directly, by use of the MAINVIEW SRM subsystem command set, or through the ISPF interface MAINVIEW SRM Global Administration panel.

SMMSYS $_{xx}$  is a required member. It must be identified on MAINVIEW SRM start up either by accepting the default value of 00 or by explicit specification. For example:

```
/S SVOS ,SUF=xx
```

All parameters for SMMSYS $_{xx}$  are on the SET statement.

## Parameter Explanations

### AOO\_SUBSYS=

**Purpose:** Specifies the AutoOPERATOR subsystems that are to receive events. You can specify up to three AutoOPERATOR subsystems.

**Syntax:** AOO\_SUBSYS=*xxxx* or AOO\_SUBSYS=(*xxxx,xxxx,xxxx*)  
where *xxxx* is the four-character subsystem name used by the AutoOPERATOR component within the OS/390 image.

**Required:** Only if routing events to AutoOPERATOR.

**Default:** None

### AUTO\_MXTSK=

**Purpose:** Specifies the maximum number of tasks to be used in collecting volume space or data set information in response to automation requests for any AUTO function

**Syntax:** AUTO\_MXTSK=*nn*  
where *nn* is a value from 2-30.

This value can be adjusted higher to decrease elapsed time of automation requests; however, this will increase the virtual storage utilization of the SVOS address space. If increased high enough, virtual storage in the SVOS address space will be completely utilized, resulting in space-related abends within SVOS processes. Care should be taken when adjusting this value.

**Required:** No

**Default:** AUTO\_MXTSK=3

### AUTOJ\_OINDX=

**Purpose:** Specifies the prefix value for the data set name that is appended to the AUTOJCL.DYHMMDD.THMMSSTT to create the skeleton JCL data set name.

**Syntax:** AUTOJ\_OINDX=*xxxxxxxxxxxxxx*  
where *xxxxxxxxxxxxxx* is 1-19 characters meeting valid data set naming convention standards

**Required:** No

Default: None

**AUTOPROC=**

Purpose: Specifies the name of the cataloged procedure used to start SG-Auto. The cataloged procedure is distributed in BBSAMP as member SGAPROC.

Syntax: `AUTOPROC=xxxxxxx`  
where `xxxxxxx` is a 1–8 character string.

Required: No

Default: None

**BBI3\_SSID=**

**Purpose:** Specifies the CAS subsystem name to which the SVOS PAS should connect. Because the MAINVIEW Infrastructure (MVI) connection occurs during SVOS startup, SVOS will not start if BBI3\_SSID is not specified. To update the value of BBI3\_SSID, SVOS must be stopped and restarted; it cannot be refreshed.

The CAS subsystem name is specified in the SSID= parameter on the PARM= keyword for the CAS JCL EXEC statement.

**Syntax:** BBI3\_SSID=xxxx

where xxxx is a 1–4 character string.

**Required:** Yes

**Default:** None

**BCDSn=**

**Purpose:** Specifies HSM CDS database files to be used by the MAINVIEW SRM Reporting component.

**Syntax:** BCDSn=xxxxxxxx

where *n* is the multi-cluster data set number (1, 2, 3, or 4)

where xxxxxxxx is a backup data set name

**Required:** No. If not specified MAINVIEW SRM will discover the HSM BCDS data set name when SVOS is started. BMC Software recommends that you let SVOS discover the data set names; however, discovery requires that the DFHSM address space be active prior to SVOS being started. If this requirement cannot be met, this keyword should be used to specify the name.

Under special circumstances it is desired to not allocate any of the HSM CDS data sets. When this is the situation, any one of the CDS files must be specified as 'NONE'. All of the CDS data sets (BCDS1, MCDS1, and OCDS) may be specified as 'NONE' but it is only necessary to specify one for this feature to take place.

**Default:** None

**BLKINPUT=**

**Purpose:** Changes block size for input data sets.

Syntax: BLKINPUT=Y/N

Required: No

Default: BLKINPUT=N

### **BLKOLDSR=**

Purpose: Changes blocksize for output data sets opened with disposition of old or shared.

Syntax: BLKOLDSR=Y/N

Required: No

Default: BLKOLDSR=N

### **CAL=**

Purpose: Specifies the suffix of the SMCALSxx member. SMCALSxx contains calendar parameters used to specify non-working days for the DFHSM-related functions.

Syntax: CAL=xx

where xx is any two-character string. A single character is not allowed.

Required: No

Default: None

### **CHECK=**

Purpose: Specifies candidate volume replacement during allocation for multivolume data sets. If multiple volumes are requested for a data set, VSAM or non-VSAM, the CHECK parameter controls whether the first volume or all volumes are immediately assigned from the assigned pool.

Syntax: CHECK=FIRST/ALLVOLS

Required: No

Default: CHECK=FIRST



**CNFG\_MXTSK=**

**Purpose:** The value indicates the maximum number of tasks to be used by the configuration component to collect configuration data. This value can be lowered to reduce the CPU and virtual storage consumption of the SVOS address space during a configuration collection, at the cost of increased response time to complete the collection process. This value can be increased to reduce the response time of a configuration collection, at the expense of CPU consumption and virtual storage utilization during a configuration collection. Increasing this value past a certain point could cause storage-related errors during configuration collection. The point at which this occurs is dependent on the private-area size of the OS/390 system as well as the REGION= parameter of the SVOS started task JCL.

**Syntax:** CNFG\_MXTSK=*nn*  
where *nn* a numeric range of 2-30

**Required:** No

**Default:** CNFG\_MXTSK=15

**CRITLIST=**

**Purpose:** Specifies the suffix of the SMCRT<sub>xx</sub> member. SMCRT<sub>xx</sub> members contain lists of critical data set names to be used by the VOLSEL option of CRITDSN.

**Syntax:** CRITLIST=*xx*

**Required:** No

**Default:** None

**DADSMEX=**

**Purpose:** Determines if the DADSM preprocessing exit (IGGPREF00) is called. If it is called and the exit returns a zero, the volume is used for a volume switch. If the exit returns a four, scanning continues for an acceptable volume. If the exit returns an eight, no volume switch occurs.

**Syntax:** DADSMEX=*Y/N*

**Required:** No

**Default:** DADSMEX=Y

**DASDGENR=**

**Purpose:** Specifies the only generic DASD names that are to be processed for specific functions if no space requirements are specified. Standard pooling functionality allows the JCL UNIT parameter to specify a pool name. If this is not desired, DASDGENR can be used to intercept data set allocations without space information. Such allocations will only be considered for subsequent processing if the unit is found in an internal table (3380, 3350, and so on) or if the unit is found in the DASDGENR list. DASDGENR affects functions DASDPOOL, DSNCHECK, SETEXPDT, and FORCECAT.

**Syntax:** DASDGENR=(xxxxxxxx,xxxxxxxx,...)

where xxxxxxxx is a generic DASD name 1–8 characters long. Multiple names can be specified.

**Required:** No

**Default:** None

**Note:** If this parameter is not coded, all data sets with DASD generics or esoterics will be processed. If any parameter is coded for DASDGENR, only unit names in the DASDGENR list will be processed, so all generic/esoteric names that are to be processed by EasyPOOL should be specified.

**DATEFMT=**

**Purpose:** Specifies the format of calendar dates in MAINVIEW SRM reports and display screens.

This parameter does not apply to the format of input dates for SMCALSxx parameters.

**Syntax:** DATEFMT=MMDD/DDMM

where MMDD specifies a date format of mm/dd/yyyy (American style), and DDMM specifies a date format of dd/mm/yyyy (European style).

**Required:** No

**Default:** DATEFMT= MMDD

**DCTYPE=**

**Purpose:** Specifies which device characteristics are to be maintained across volumes during a volume switch. For example, if your installation wishes to segregate data sets residing on cached and non-cached devices, the DCTYPE=(CACHE) operand can be used to ensure that the cache property is maintained during a volume switch. Data sets residing on a cached device only switch to volumes that are also cached. Likewise, data sets on non-cached devices switch to only non-cached volumes.

**Syntax:** DCTYPE=(CACHE,SHARE, DUALCOPY, FASTWRITE)

Multiple device characteristics can be specified.

If DCTYPE has been specified, CHECK=DC on the SKIP parameter can be used to ignore the device characteristic checks for special conditions.

**Required:** No

**Default:** None

**DFREORGPRC=**

**Purpose:** Defines default SPACVOLA reorganize procedure name of the procedure that is started if SPACVOLA reorganize processing is requested.

**Syntax:** DFREORGPRC=xxxxxxxx

where xxxxxxxx is the name of the reorganize procedure (1-8 characters).

**Required:** No

**Default:** DFREORGPRC=REORGPRC

**DIAG=**

**Purpose:** Suffix of parameter member SMDIAGxx.

**Syntax:** DIAG=xx

where xx is any two-character string. A single character is not allowed.

**Required:** No

**Default:** None

**DIAGMSDD=**

Purpose: Establishes WTO message tracing.

Syntax: `DIAGMSDD=xxxxxxx`  
where `xxxxxxx` is a 1–8 character DD name.

Required: No

Default: `DIAGMSDD=PRO$D$N$`

**DISPLAY=**

Purpose: Determines which functions are displayed in the ISPF interface functions panel. `DISPLAY=ALL` displays all functions for the MAINVIEW SRM products; `DISPLAY=LIC` shows only those functions in the products for which you are licensed; `DISPLAY=ACT` shows only those functions that are active.

Syntax: `DISPLAY=ALL/LIC/ACT`

Required: No

Default: None

**DMYUNIT=**

Purpose: Defines the conversion of a nonexistent UNIT parameter to a valid UNIT parameter only if `JCLEXT=Y`.

Syntax: `DMYUNIT=(xxxxxxx,zzzzzzzz,...xxxxxxx,zzzzzzzz)`  
where the first `xxxxxxx` is the invalid UNIT parameter to be converted to the valid UNIT parameter `zzzzzzzz`. Multiple pairs of DMYUNITs can be specified.

Required: No

Default: None

**DP\_RENAME=**

**Purpose:** Specifies that during DADSM RENAME, DASDPOOL will be driven to verify that the POOL containing the volume on which the data set currently resides is also a POOL that would be assigned to the renamed data set.

When DP\_RENAME=Y, if the first POOL in which the current volume is found does not match a POOL that would be assigned to the renamed data set, the RENAME will be denied.

When DP\_RENAME=A, if any POOL in which the current volume is found does not match a POOL that would be assigned to the renamed data set, the RENAME will be denied.

**Syntax:** DP\_RENAME=Y/N/A

**Required:** No

**Default:** DP\_RENAME=N

The new FLST/RLST parameter DADSM\_FUNC should be used to limit the data sets processed by enabling this option.

**DUMPDD=**

**Purpose:** Produces SYS1.DUMPxx dump if MAINVIEW SRM abends.

**Syntax:** DUMPDD=xxxxxxxx

where xxxxxxxx is a 1–8 character DD name.

**Required:** No

**Default:** DUMPDD=PRODUMP

**ETS\_ID=**

ETS\_ID is no longer supported. The Ensign Alarm Console is no longer supported in the Automation component. If the keyword is specified, the SVM0766I messages is issued and the value is ignored.

**EVNT=**

- Purpose:** Specifies the suffix you assign to the name of the SMEVNT<sub>xx</sub> event definition member.
- Syntax:** EVNT=<sub>xx</sub>  
where <sub>xx</sub> is the two-character suffix of the SMEVNT<sub>xx</sub> member name.
- Required:** Yes
- Default:** None

**FDRIAM=**

- Purpose:** Determines whether a data set is an IAM data set.  
**Note:** *Only* IAM customers should set this parameter to FDRIAM=Y.
- Syntax:** FDRIAM=Y/N
- Required:** No
- Default:** FDRIAM=N

**FORPLEXNAME=**

- Purpose:** Specifies one or more user-defined PLEXNAMEs that can be included or excluded in a sysplex environment.
- Syntax:** FORPLEXNAME=<sub>xxxxxxxx</sub>  
where <sub>xxxxxxxx</sub> is 1-8 characters
- Required:** No
- Default:** FORPLEXNAME=*current image*

**FORSMFID=**

**Purpose:** Specifies SMF records that can be included or excluded in a sysplex environment.

**Syntax:** FORSMFID=xxxxxxxx  
where xxxxxxxx is 1-8 characters

**Required:** No

**Default:** FORSMFID=*current image*

**FORSYSID=**

**Purpose:** Specifies user-defined system IDs that can be included or excluded in a sysplex environment.

**Syntax:** FORSYSID=xxxxxxxx  
where xxxxxxxx is 1-8 characters

**Required:** No

**Default:** FORSYSID=*current image*

**FUNC=**

**Purpose:** Specifies the suffix of the SMFUNCxx member. SMFUNCxx contains function definition parameters. A function must be included in the SMFUNCxx member to be available during MAINVIEW SRM execution. Each function definition identifies two other members that define

- the resources affected by the function (SMFLSTxx—the filter list)
- how those resources are affected (SMRLSTxx—the rules list)

Note that some functions do not use a rules list, but all functions require a filter list. See the discussion for SMFUNCxx in “Function Member Parameters” on page D-107.

**Syntax:** FUNC=xx  
where xx is any two-character string. A single character is not allowed.

**Required:** Yes

**Default:** None

**HISTDAYS=**

**Purpose:** Specifies the number of days of data to retrieve from the performance collector and temporarily store in a data space. Values are 0-14 (7 or 14 recommended), which indicate the number of days of data to be retrieved. The default value is 0, which indicates that the data space is created, but no historical performance data is gathered and no HISTDPO pooling can be performed. If this value is changed, the historical performance data collector (SVSGP) must be stopped and restarted.

**Syntax:** HISTDAYS=*nn*

**Required:** No

**Default:** HISTDAYS=0

**HLOGAUTH=**

**Purpose:** Specifies the *hours* component of the duration between automatic logfile switching.

The Reporting component extracts records from the DFHSM logfiles. If HLOGCOLL=Y is specified, MAINVIEW SRM will switch the DFHSM logfile and extract the required records automatically. The switching interval is specified in hours and minutes by the parameters HLOGAUTH and HLOGAUTM.

**Syntax:** HLOGAUTH=*nn*

where *nn* specifies a number of hours in the range 0–24.

**Required:** Yes

**Default:** HLOGAUTH=00



**HLOGAUTM=**

**Purpose:** Specifies the *minutes* component of the duration between automatic logfile switching.

The Reporting component extracts records from the DFHSM logfiles. If HLOGCOLL=Y is specified, the system will switch the DFHSM logfile and extract the required records automatically. The switching interval is specified in hours and minutes by the parameters HLOGAUTH and HLOGAUTM.

**Syntax:** HLOGAUTM=*nn*

where *nn* specifies a number of minutes in the range 0–60.

**Required:** No

**Default:** HLOGAUTM=00

**HLOGCOLL=**

**Purpose:** Specifies whether MAINVIEW SRM will perform DFHSM logfile switching and record extraction automatically.

The Reporting component requires certain records from the DFHSM logfiles. If HLOGCOLL=Y is specified, MAINVIEW SRM will switch the DFHSM logfiles and run a record extraction program automatically at the interval specified by the HLOGAUTH/M parameters. For more information on DFHSM logfile switching and extraction, see the *MAINVIEW SRM Reporting Reference Guide*.

**Syntax:** HLOGCOLL=Y/N

**Required:** No

**Default:** HLOGCOLL=N

**HLOGINDX=**

**Purpose:** Specifies the prefix of the data set that will contain the records extracted from the DFHSM logfile. The prefix can contain up to 20 characters in any number of name qualifiers. The full data set name generated for the log extract file is

*prefix.Dyymmdd.Thhmmss.SYSsystem-id*

**Syntax:** HLOGINDX=xxxxxxxxxxxxxxxxxxxx

**Required:** A name is required for the HSM collector reporting facility to be functional.

Default: None

**HLOGLIM=**

**Purpose:** Specifies a limit to the number of log extract data sets that will be allocated for one report request. By default, the system calculates the number of log extract data sets that it will allocate based upon how many active allocations reside in the SVOS address space. HLOGLIM allows users to limit the number of log extracts that get allocated for one request. Consequently, one request will not use up all of the available allocation openings, which would cause other requests to prematurely end without gathering of all the requested data. Additionally, using fewer allocation openings lowers the amount of I/O that is used by the SVOS address space.

**Syntax:** HLOGLIM=*xxxx*  
where *xxxx* is a number in the range of 0–3000

**Required:** No

**Default:** none

**HLOGPRIM=**

**Purpose:** Specifies the number of tracks to be allocated as the primary extent amount for the log extract file. One-half of the primary extent is allocated for the secondary (with a minimum of 1). If not specified, 15 tracks are used for primary and 10 tracks for secondary.

**Syntax:** HLOGPRIM=*nnnn*  
where *nnnn* is a number in the range 1–9999.

**Required:** No

**Default:** HLOGPRIM=15

**HLOGTASK=**

**Purpose:** Specifies the name of the procedure to be run following the DFHSM logfile switch program execution. If HLOGCOLL=Y is specified MAINVIEW SRM will execute the DFHSM logfile switch program.

HLOGTASK may be used to run a procedure associated with the logfile switch performed by the MAINVIEW SRM utility.

**Note:** When LOGY files are processed during switching, the HLOGTASK procedure is executed after the switching is complete. If you do not want the procedure submitted, blank out the value for HLOGTASK or code as comment or remove the line in the startup parameters.

**Syntax:** HLOGTASK=xxxxxxx  
where xxxxxxx is 1-8 characters

**Required:** No

**Default:** None

**HLOGUNIT=**

**Purpose:** Specifies the esoteric or generic unit name for the allocation of the log extract file. If not specified, SYSALLDA is used.

**Syntax:** HLOGUNIT=xxxxxxx  
where xxxxxxx is 1-8 characters

**Required:** No

**Default:** HLOGUNIT=SYSALLDA

**HLOGYDSN=**

**Purpose:** Specifies the fully-qualified data set name of DFHSM logfile Y.

**Syntax:** HLOGYDSN=xxxxxxxx.xxxxxxx....  
where xxxxxxx is 1-8 characters

**Required:** No. If not specified MAINVIEW SRM will discover the HSM LOG Y data set name when SVOS is started. BMC Software recommends that you let SVOS discover the data set names; however, discovery requires that the DFHSM address space be active prior to SVOS being started. If this requirement cannot be met, this keyword should be used to specify the name.

Default: None

**HSMACTID=**

Purpose: Specifies the high-level data set name qualifier for the DFHSM activity data sets. This name qualifier is used by the output management facility to build the name of the DFHSM activity data sets that are used as input.

Syntax: HSMACTID=xxxxxxx  
where xxxxxxx is any 1–8 character string.

Required: No

Default: HSMACTID=DFHSM

**IGNOREDD=**

Purpose: Suppresses all MAINVIEW SRM activity for the jobstep containing the specified DD name. No MAINVIEW SRM functions will occur for any data set in the jobstep. Note that the IGNORE parameter in the DIAG member will override the DD statement's presence.

Syntax: IGNOREDD=xxxxxxx  
where xxxxxxx is 1–8 character DD name.

Required: No

Default: IGNOREDD=PROIGN

**JCLEXT=**

**Purpose:** Specifies if EasyPOOL will obtain volume and unit information after MVS accessed the catalog.

**Syntax:** JCLEXT=Y/N

**Required:** No

**Default:** JCLEXT=Y

**Note:** If JCLEXT=Y is specified, all non-valid unit names must be specified in DMYUNIT; otherwise, MVS will fail the allocation. Also, JCLEXT=Y should be used carefully if PROCOLD=Y is also specified, because JCLEXT will find a unit and volume from the catalog, whereas PROCOLD=Y will allow the existing data set to be reprocessed, possibly assigning a different (and invalid) volume.

To use the UNIT= filter for VTS data sets, you must specify JCLEXT=N.

**JCLUREQ=**

**Purpose:** When PROCOLD=Y is specified and EasyPOOL is analyzing a DD statement for an existing data set, the JCLUREQ parameter determines if UNIT information is required. If JCLUREQ=Y is specified, MAINVIEW SRM intercepts the DD statement only when the JCL specifies UNIT parameter. This allows the installation to correct JCL that uses an invalid unit parameter for existing data sets without analyzing DD statements that are correctly specified. EasyPOOL examines every DD statement associated with an existing data set if you specify JCLUREQ=N.

**Note:** JCLUREQ=N needs to be specified when processing UNIT=AFF groups because the unit field in the SIOT consists of blanks.

**Syntax:** JCLUREQ=Y/N

**Required:** No

**Default:** JCLUREQ=N

**MAXVOL=**

**Purpose:** Limits the number of volumes that SPACVOLA allows a data set to use. When additional units are allocated with UNIT=(SYSDA,*n*), they are counted if space is obtained on a volume. If *n* is greater than the MAXVOL value, SPACVOLA does not limit the use of the additional volumes, but does not add additional volumes through a volume switch.

**Syntax:** MAXVOL=*nn*

where *nn* is any number in the range 1–59.

**Required:** No

**Default:** MAXVOL=5

**Note:** This option can be set globally and overridden by individual SPACVOLA RLST action statements.

The MAXVOL parameter and MVS will allow a data set to extend across as many as 59 volumes. However, some products using DFP 3.3 CAMLST services will only process up to 20 volumes, due to a limitation in the CAMLST processing (see the IBM manual SC26-4567 *MVS/DFP 3.3 System Programming Reference*, section 4.3, Retrieving Information from a Catalog). If you are using DFP 3.3 or earlier, and are using products that use CAMLST services to process multivolume data sets, consider setting the MAXVOL parameter to less than 20 volumes.

**MCDS $n$ =**

**Purpose:** Specifies HSM migrated data set file allocated during system startup.

**Syntax:** MCDS $n$ =xxxxxxxx

where  $n$  is the multi-cluster data set number (1, 2, 3, or 4)

where xxxxxxxx is a migrated data set name

**Required:** No. If not specified MAINVIEW SRM will discover the HSM MCDS data set name when SVOS is started. BMC Software recommends that you let SVOS discover the data set names; however, discovery requires that the DFHSM address space be active prior to SVOS being started. If this requirement cannot be met, this keyword should be used to specify the name.

Under special circumstances it is desired to not allocate any of the HSM CDS data sets. When this is the situation, any one of the CDS files must be specified as 'NONE'. All of the CDS data sets (BCDS1, MCDS1, and OCDS) may be specified as 'NONE', but it is only necessary to specify one for this feature to take place.

**Default:** None

**MODTRCDD=**

**Purpose:** Establishes module entry/exit tracing.

**Syntax:** MODTRCDD=xxxxxxxx

where xxxxxxxx is a 1–8 character DD name.

**Required:** No

**Default:** MODTRCDD=PROTRCE

**MREDUCE=**

**Purpose:** Determines whether secondary space reduction can occur on multivolume data sets that were allocated with JCL. For example, MREDUCE=N would not allow secondary space reduction for the first three volumes when allocated with UNIT=(SYSDA,4).

**Syntax:** MREDUCE=Y/N

**Required:** No

**Default:** MREDUCE=Y

**MSGID=**

**Purpose:** Specifies the inclusion of the MAINVIEW SRM message identifier in the message text. For example:

MSGID=Y

```
15.00.30 JOB29640 SVM3352I
EMPCRMX,SA,DD1,EMPCRM.Q2.TEST REQUESTED SPACE
0463KB EXCEEDS LIMIT
```

MSGID=N

```
14.59.37 JOB29639      EMPCRMX,SA,DD1,EMPCRM.Q2.TEST
REQUESTED SPACE 0463KB EXCEEDS LIMIT
```

**Syntax:** MSGID=Y/N

**Required:** No

**Default:** MSGID=Y

**MSGLVL=**

**Purpose:** Specifies the level of messages that will be printed. The options are I=Informational, W=Warning, E=Error, S=Severe error. Messages are inclusive of increasing levels of severity. When a particular level is chosen, messages of that level and greater will be printed. For example, if E is chosen, messages of severity E or S are printed.

**Syntax:** MSGLVL=I/W/E/S

**Required:** No

**Default:** MSGLVL=I

**MSGPREF=**

**Purpose:** Specifies the three-character message identifier prefix on MAINVIEW SRM messages.

**Syntax:** MSGPREF=xxx/SVM

where xxx is any three-character string. Fewer than three characters is not allowed.

**Required:** No



Default: MSGPREF=SVM

NOCATDYN=

Purpose: Allows NOCATLG2 to process dynamically allocated data sets.

Syntax: NOCATDYN=Y/N

Required: No

Default: NOCATDYN=N

**NOCATPFX=**

Purpose: Specifies the second-level qualifier to be used by NOCATLG2 when renaming a data set. NOCATLG2 can be directed to rename existing data sets by specifying the NOCATLG2=RENAME operand on the RLST action parameter.

Syntax: NOCATPFX=xxx

where xxx is 1 to 3 alphanumeric or national characters. The first character must be alphabetic.

Required: No

Default: NOCATPFX=BAB

**NOCATPRG=**

Purpose: Specifies if NOCATLG2 can scratch a data set that has not reached its expiration date when NOCATLG2=SCRATCH is specified in the RLST action parameter. NOCATPRG=Y indicates that the data set is to be scratched regardless of expiration date (that is, scratch is issued with the PURGE option). This option can be set globally and overridden on individual action statements by the PURGE operand on the RLST action parameter.

Syntax: NOCATPRG=Y/N

Required: No

Default: NOCATPRG=N

**NOCATSEC=**

**Purpose:** Specifies the level of security checking that NOCATLG2 performs before scratching or renaming a data set. If the creator of the new data set does not have the specified level of authority, NOCATLG2 will not scratch or rename the existing data set.

**Syntax:** NOCATSEC=NONE/CREATE/READ/UPDATE/ALTER

**Required:** No

**Default:** NOCATSEC=NONE

**NOCATSMS=**

**Purpose:** Specifies whether SMS-managed data sets can be renamed, uncataloged, or scratched by NOCATLG2. If NOCATSMS=Y is specified, NOCATLG2 renames, uncatalogs, or scratches an existing SMS-managed data set if NOCATLG2=RENAME, UNCATLG, or SCRATCH is specified. The existing SMS-managed data set is renamed/uncataloged/scratched whether the new data set to be allocated is SMS-managed.

If NOCATLG2=UNCATLG is specified and the existing data set is SMS-managed, NOCATLG2 issues a DELETE NOSCRATCH to remove the catalog entry for the data set. The data set will exist on the SMS-managed volume but will not be cataloged. The catalog entry will point to the newly allocated data set instead.

**Syntax:** NOCATSMS=Y/N

**Required:** No

**Default:** NOCATSMS=N

**NOCATTIM=**

**Purpose:** Specifies the amount of time (in minutes) that StopX37/II will wait while attempting to perform NOCATLG2 processing (DELETE option only) on an SMS managed data set. If NOCATLG2 runs while backups or restores are running, it is possible that NOCATLG2 processing will be delayed. While processing a data set, NOCATLG2 will time out after the time specified in global NOCATTIM has expired and therefore will not complete processing. NOCATTIM allows this time out value to be changed. If you are seeing the following message in your job log:

SMS4105I UNABLE TO SCRATCH/RENAME; REASON = UNABLE TO COMPLETE FUNCTION WITHIN 5 MINUTES

then you can try increasing the value specified in NOCATTIM. The optimal solution however would be to re-schedule the job during a period when backups or restores are not running.

Should you chose to increase the value in NOCATTIM, please be advised that the StopX37/II loader will not shut down until all StopX37/II/EasyPOOL work has completed. Therefore, increasing the value of NOCATTIM could increase the time it takes for the loader to shut down.

**Syntax:** NOCATTIM=*nn*

where *nn* specifies the number of minutes in the range of 1-59.

**Required:** No

**Default:** NOCATTIM=05

**NOCATVOL=**

**Purpose:** Allows a new data set to be allocated to the same volume to which it was previously cataloged. This can occur when a catalog entry is *orphaned* because a data set is removed from a volume without the data set being uncataloged or when a catalog entry is added for a data set but the data set is never actually created. This operand acts differently depending on the NOCATWHEN setting in the NOCATLG2 RLST. If NOCATWHEN=TERM is specified and NOCATLG2=UNCATLG is specified in the relevant RLST action parameter, NOCATVOL=SAME indicates that the data set should be uncataloged and recataloged even if the old and new volumes are the same and NOCATVOL=DIFF means that the old catalog entry is left in the catalog if the old and new volumes are the same.

If the relevant RLST action parameter specifies NOCATLG2=SCRATCH or NOCATLG2=RENAME, NOCATVOL=DIFF is forced and the NOCATLG2 processing will fail with an error message if the old and new volumes are the same.

If NOCATWHEN=ALLOC is specified, NOCATVOL is ignored and NOCATLG2 processing occurs, even if the old and new volumes are the same.

Syntax: NOCATVOL=SAME/DIFF

Required: No

Default: NOCATVOL=DIFF

### **NOCATWHEN=**

Purpose: Specifies when NOCATGL2 processing occurs for non-SMS managed data sets. ALLOC indicates that NOCATGL2 processing occurs during data set allocation. TERM indicates that NOCATLG2 processing occurs during step termination. NOCATLG2 processing for SMS-managed data sets must occur during data set allocation.

Syntax: NOCATWHEN=ALLOC/TERM

Required: No

Default: NOCATWHEN=ALLOC

**Note:** When NOCATWHEN=ALLOC and no volser is specified, you will not be able to filter on parameter VOL=. To be able to filter on VOL=, you must specify NOCATWHEN=TERM.

### **OCDS=**

Purpose: Specifies HSM OCDS data sets to be defined and allocated during system startup.

Syntax: OCDS=xxxxxxx

*where xxxxxxxx is an OCDS data set name*

Required: No. If not specified MAINVIEW SRM will discover the HSM OCDS data set name when SVOS is started. BMC Software recommends that you let SVOS discover the data set names; however, discovery requires that the DFHSM address space be active prior to SVOS being started. If this requirement cannot be met, this keyword should be used to specify the name.

Under special circumstances it is desired to not allocate any of the HSM CDS data sets. When this is the situation, any one of the CDS files must be specified as 'NONE'. All of the CDS data sets (BCDS1, MCDS1, and OCDS) may be specified as 'NONE' but it is only necessary to specify one for this feature to take place. In this situation some views will not be available.

Default: None

### **OPMHLQ=**

Purpose: Specifies the high-level qualifier for the data sets created by the Output Management Facility, which allows selection of DFHSM and DFDSS messages for reporting and analysis.

Syntax: OPMHLQ=xxxxxxxx

where xxxxxxxx is any 1–8 character string.

Required: Only if the MAINVIEW SRM Output Management Facility is to be used.

Default: None

### **ORIGDATA=**

Purpose: During EasyPOOL processing, ORIGDATA specifies whether VOL and UNIT contain the original volser and unit values from the JCL or the current value. If ORIGDATA=PRO is used, the selection fields VOL and UNIT will always contain the original volser and unit from the JCL. This is how these fields are handled in earlier releases of MAINVIEW SRM. In POOL-DASD these fields reflected any changes made to the volser and unit by showing the current value. If ORIGDATA=POOL is used, MAINVIEW SRM will reflect the current values for these fields.

**Note:** The fields ORIGVOL and ORIGUNIT will always contain the original VOLSER and UNIT that were specified in the JCL.

Syntax: ORIGDATA=PRO/POOL

Required: No

Default: ORIGDATA=PRO

### **PASSWORD=**

Purpose: Specifies the passwords supplied by BMC Software. One or more passwords can be required, depending on which MAINVIEW SRM components you purchased.

Syntax:                   PASSWORD=xxxxxxxx

where xxxxxxxx is a 16-character string.

Required:                Yes

Default:                 None

**PERFRM\_PRC=**

Purpose:                   Specifies the name of the procedure used to start the historical performance collector. The procedure is distributed in *?prefix.BBSAMP* as member SGPPROC.

Syntax:                   PERFRM\_PRC=xxxxxxx

where xxxxxxx is an 8-character string.

Required:                No

Default:                 None

**POOL=**

Purpose:                   Specifies the suffix of an SMPOOL<sub>xx</sub> member. SMPOOL<sub>xx</sub> contains non-SMS managed device pool definition parameters. It names pools and assigns volumes to pools.

Syntax:                   POOL=xx

where xx is any two-character string. A single character is not allowed.

Required:                Yes

Default:                 None

**PROCOLD=**

**Purpose:** Specifies if EasyPOOL will intercept DD statements that specify OLD allocations. EasyPOOL always intercepts NEW and MOD allocations.

EasyPOOL also intercepts DD statements that specify the UNIT parameter when PROCOLD=Y is specified, which is useful for correcting questionable JCL. For example, assume DASDPOOL pools a data set to a TEST volume. If a later step wants to use the data set and specifies UNIT=PROD, the job receives a JCL error. You can correct this situation by specifying PROCOLD=Y.

**Syntax:** PROCOLD=Y/N

**Required:** No

**Default:** PROCOLD=N

**Note:** PROCOLD=Y is useful to override volsers that are hardcoded in JCL for old data sets. PROCOLD=Y will logically remove that hard-coded volume. However, use caution if you also specify JCLEXT=Y, which will assign a unit and volume from the catalog.

**REJECT=**

**Purpose:** Controls termination of processing at the first or last data set rejected by REJECT=Y in DASDPOOL or DSNCHECK. If termination is to take place on the first rejected data set, code REJECT=FIRST. If all data sets are to be processed before control is returned to MVS allocation, code REJECT=LAST.

**Syntax:** REJECT=FIRST/LAST

**Required:** No

**Default:** REJECT=FIRST

**REQTYPE=**

**Purpose:** Specifies whether the MNTYPE statement in SPACVOLA is considered the request type instead of the mount type. For example, VOL=SER=WORK01 can be considered a private request even though the pack was mounted storage. MNTYPE defaults to the request type.

**Syntax:** REQTYPE=Y/N

**Required:** No

Default: REQTYPE=Y

**RLS=**

Purpose: Specifies whether the system should determine the VSAM Record Level Sharing environment available for DFHSM.

Syntax: RLS=Y/N

Required: N

Default RLS=N

**SCAT=**

Purpose: Forces a catalog update to occur during the volume switch. By default, when SPACVOLA performs a volume switch on a permanent data set, the catalog is updated to contain the new volumes at step termination. For SMS-managed data sets, the catalog is always updated immediately.

Syntax: SCAT=STEPEND/IMMEDIATE

Required: No

Default: SCAT=STEPEND

**SG\_INITPOOL=**

Purpose: Specifies the maximum number of defined pools included in a single snapshot. The maximum number of defined pools at initialization of a new linear data set is 3,995 unless a greater value is specified on this parameter. After initialization, data is collected into a snapshot for the number of pools specified on this parameter.

Syntax: SG\_INITPOOL=*nnnnnn*  
where *nnnnnn* is a value in the range 10–999999

Required: No

Default: 1000

**Note:** Do not modify the default value unless you must.



**SG\_INITVOL=**

**Purpose:** Specifies the maximum number of defined volumes included in a single snapshot. The maximum number of defined volumes at initialization of a new linear data set is 6,625 unless a greater value is specified on this parameter. After initialization, data is collected into a snapshot for the number of volumes specified on this parameter.

**Syntax:** SG\_INITVOL=*nnnnnn*  
where *nnnnnn* is a value in the range 10–999999

**Required:** No

**Default:** 3000

**Note:** Do not modify the default value unless you must.

**SG\_IXFPNTVL=**

**Purpose:** Specifies the number of hours between refreshes of the IXFP data tables

**Syntax:** SG\_IXFPNTVL=*nn*

**Required:** No

**Default:** None

**SG\_MAXACCT=**

**Purpose:** Specifies the maximum number of active accounts in the application database.

**Syntax:** SG\_MAXACCT=*nnnnn*  
where *nnnnn* is a value in the range 328–32765

**Required:** No

**Default:** Extracted from the application database

**SG\_MAXPOOL=**

**Purpose:** Specifies the number of pools that can be assigned to a volume. Used by the data collector when building pool snapshots.

**Syntax:** SGMAXPOOL=*n*

where  $n$  is a value in the range 1–8

Required: No

Default: SGMAXPOOL=1

### **SG\_MAXSSDSZ=**

Purpose: Specifies the number of cylinders used for a solid state disk drive. Used to distinguish between emulated and real DASD. Any device that does not exceed the value specified on this parameter is considered a solid state device.

Syntax: SG\_MAXSSDSZ=*nnnnn*

where *nnnnn* is a value less than 32766

Required: No

Default: SG\_MAXSSDSZ=0

### **SG\_READNTVL=**

Purpose: Specifies the frequency (in minutes) at which the space collector scans the DASD volumes for historical space information to create a snapshot in memory.

Syntax: SG\_READNTVL=*nnnn*

where *nnnn* is a value in the range 5–9999

Required: No

Default: SG\_READNTVL=30

### **SG\_RETRYLIM=**

Purpose: Specifies the number of abend conditions the data collector should ignore.

Syntax: SG\_RETRYLIM=*nnnn*

where *nnnn* is a value from 5–9999

Required: No

Default: SG\_RETRYLIM=10

**SG\_SIBSTK=**

**Purpose:** Specifies the IXFP SIBBATCH parameter member to be used by the MAINVIEW SRM IXFP services for communications with the IXFP address space. This parameter must be specified to activate RVA collection in the space collector. Without this parameter, the space collector views will have a physical disk ID and box serial number of question marks.

**Note:** If you have RVA, but do *not* want recording turned on, you must delete the SG\_SIBSTK variable or make it empty.

**Syntax:** SG\_SIBSTK=xxxxxxx  
where xxxxxxx is 2-8 characters

**Required:** No

**Default:** None

**SG\_SPACHLDR=**

**Purpose:** Defines a data set name mask that the space collector can use to identify space holder data sets.

**Syntax:** SG\_SPACHLDR=MASK  
where MASK is a space holder data set name mask

**Required:** No

**Default:** None

**SG\_SUBTASKS=**

**Purpose:** Defines the number of volumes that can be read in parallel.

**Syntax:** SG\_SUBTASKS=nn  
where nn is a value in the range 2–10

**Required:** No

**Default:** SG\_SUBTASKS=3

**SG\_VVDSINFO=**

Purpose: Indicates whether the VVDS size and percentage used should be calculated for each volume processed by space collector.

Syntax: SG\_VVDSINFO=YES/NO  
  
Specifying YES can have a performance implication due to the volume VVDS being allocated, read, and deallocated at each collection interval.

Required: No

Default: SG\_VVDSINFO=NO

**SG\_WRITNTVL=**

Purpose: Specifies the frequency (in minutes) at which snapshots are written to the space database.

Syntax: SG\_WRITNTVL=*nn*  
  
where *nn* is a value in the range 1–1439

Required: No

Default: SG\_WRITNTVL=30

**SGA\_ENQSCOP=**

Purpose: Specifies the operational environment in which SG-Auto is to run. If GLOBAL is specified, SG-Auto issues an ENQ with the SYSTEMS parameter. If LOCAL is specified, SG-Auto issues an ENQ with the SYSTEM parameter. Refer to the appropriate IBM documentation for a description of the ENQ macro options.

Syntax: SGA\_ENQSCOP=GLOBAL/LOCAL

Required: Required for SG-Auto

Default: SGA\_ENQSCOP=GLOBAL

**SGACMD=**

Purpose: Specifies the two-position suffix of the initial command for executing the SG-Auto started task. The suffix will be appended to SGACMD to form the member name as it exists in the MAINVIEW SRM parmlib.

Syntax: SGACMD=

Required: No

Default: None

**SGASCAN=**

Purpose: Specifies whether SG-Auto should be started in scan mode.

Syntax: SGASCAN=Y/N

Required: Yes

Default: None

**SGASIM=**

Purpose: Specifies whether SG-Auto should be started in SIMULATION mode.

Syntax: SGASIM=Y/N

Required: No

Default: None

**SGC\_ADDEXIT=**

Purpose: Defines the name of application Add Exit.

Syntax: SGC\_ADDEXIT=xxxxxxx

where xxxxxxx is a 1–8 character string

Required: No

Default: None

**SGC\_CHKEXIT=**

Purpose: Defines the name of application Check Exit.

Syntax: SG\_CHKEXIT=xxxxxxx

where xxxxxxx is a 1–8 character string

Required: No

Default: None

**SGC\_DEFEXIT=**

Purpose: Defines the name of application Default Exit.

Syntax: SGC\_DEFEXIT=xxxxxxx  
where xxxxxxx is a 1–8 character string

Required: No

Default: None

**SGC\_KEYEXIT=**

Purpose: Defines the name of Account Code Build Exit.

Syntax: SGC\_KEYEXIT=xxxxxxx  
where xxxxxxx is a 1–8 character string

Required: No

Default: None

**SGC\_SECEXIT=**

Purpose: Defines the name of application Security Exit.

Syntax: SGC\_SECEXIT=xxxxxxx  
where xxxxxxx is a 1–8 character string

Required: No

Default: None

**SGC\_SELEXIT=**

Purpose: Defines the name of application Select Exit.

Syntax: SGC\_SELEXIT=xxxxxxx  
where xxxxxxx is a 1–8 character string

Required: No

Default: None

**SGC\_STOGRP=**

**Purpose:** Specifies whether to retrieve SMS storage group information. This parameter can only be set to YES if SMS storage group information is required for FLST or RLST processing.

**Syntax:** SGC\_STOGRP=Y/N

**Required:** No

**Default:** SGC\_STOGRP=N

**SGC\_STORCLS=**

**Purpose:** Specifies whether to retrieve storage class information. This parameter can only be set to YES if SMS storage class information is required for FLST or RLST processing. SGC\_STORCLS includes data class, management class, and storage class information.

**Syntax:** SGC\_STORCLS=Y/N

**Required:** No

**Default:** SGC\_STORCLS=N

**SGCDSN=**

**Purpose:** Specifies the data set name for the dynamic allocation/deallocation of application database DD, namely SGCDDB. The dynamic allocation occurs with the start of the SVSGC component and deallocation occurs with the stop of SVSGC. This parameter is used only if the SGCDDB DD statement is not present with the SVOS startup JCL.

**Syntax:** SGCDSN=xxxxxx...xxxxxx

where xxxxxx...xxxxxx is a 1 to 44 character string for name of data set

**Required:** No

**Default:** None

**SGD\_PROCNM=**

**Purpose:** Specifies the name of the space collector started task.

**Syntax:** SGD\_PROCNM=SGDCOLLS

**Required:** Required to run the data collector

Default: SGD\_PROCNM=SGDCOLLS

### **SGD\_SMFID=**

Purpose: Controls the generation of SMF records for the space collector.

Syntax: SGD\_SMFID=*nnn*  
where *nnn* is a value in the range 0–255

Required: No

Default: SGD\_SMFID=0

### **SGDCOLLECT=**

Purpose: Specifies if the space collector will collect pool data. This parameter can be overridden at the pool level.

Syntax: SGDCOLLECT=Y/N

Required: No

Default: SGDCOLLECT=N

### **SGDCOLLECT*n*=**

Purpose: Specifies whether pool data will be collected in an alternate space collector. The alternate data collector to be used is identified by the suffix of *n*. This parameter can be overridden at the pool level.

Syntax: SGDCOLLECT*n*=Y/N  
where *n* is a value in the range of 1–8

Required: No

Default: None

### **SGDPROCNM*n*=**

Purpose: Specifies the cataloged procedure to be started for a specified copy of the space collector. The name of the procedure must be unique. Make sure that each procedure resides in a separate set of linear data sets (SGRDPOOL, SGRDVOL, and so on).

Syntax: SGDPROCNM*n*=xxxxxxxx



where  $n$  is a value in the range of 1–8 and *xxxxxxx* is the procedure name

Required: No

Default: None (For example, if SGDPROCNM4 is not defined, you will not be unable to issue the S SVSGD4 SVOS command.)

### **SGDSMFID $n$ =**

Purpose: Specifies the SMF record number for MAINVIEW SRM audit records written to the SMF data set for a specified copy of the space collector. (Note that SMF message generation is also controlled by the SMF parameter on individual MAINVIEW SRM functions in member SMFUNC $xx$  and by the SMF parameter on filter list specifications.)

Syntax: SGDSMFID $n$ = $nnn$

where  $n$  is a value in the range of 1–8 and where  $nnn$  is a number in the range 128–255. A zero can be specified to deactivate SMF recording.

Required: No

Default: None

### **SGINITPOOL $n$ =**

Purpose: Specifies the maximum number of defined pools included in a single snapshot for a specified copy of the space collector. The maximum number of defined pools at initialization of a new linear data set is 3,995 unless a greater value is specified on this parameter. After initialization, data is collected into a snapshot for the number of pools specified on this parameter.

Syntax: SGINITPOOL $n$ = $nnnnnnn$

where  $n$  is a value in the range of 1–8 and where  $nnnnnnn$  is a value in the range 10–999999

Required: No

Default: None

**SGINITVOL $n$ =**

**Purpose:** Specifies the maximum number of defined volumes included in a single snapshot for a specified copy of the space collector. The maximum number of defined volumes at initialization of a new linear data set is 6,625 unless a greater value is specified on this parameter. After initialization, data is collected into a snapshot for the number of volumes specified on this parameter.

**Syntax:** SG\_INITVOL=*nnnnnn*  
where *nnnnnn* is a value in the range 10–999999

**Syntax:** SGINITVOL $n$ =*nnnnnn*  
where  $n$  is a value in the range of 1–8 and where *nnnnnn* is a value in the range 10–999999

**Required:** No

**Default:** None

**SGMAXACCT $n$ =**

**Purpose:** Specifies the maximum number of active accounts in the application database.

**Syntax:** SGMAXACCT $n$ =*nnnnn*  
where  $n$  is a value in the range of 1–8 and where *nnnnn* is a value in the range 328–32765

**Required:** No

**Default:** Extracted from the application database

**SGMAXPOOL $n$ =**

**Purpose:** Specifies the number of pools that can be assigned to a volume for the specified copy of the space collector. Used by the data collector when building pool snapshots.

**Syntax:** SGMAXPOOL $n$ = $n$   
where  $n$  is a value in the range of 1–8 and  $n$  is a value in the range 1–8

**Required:** No

Default: None

**SGMAXSSDSZ $n$ =**

**Purpose:** Specifies the number of cylinders used for a solid state disk drive for a specified copy of the space collector. Used to distinguish between emulated and real DASD. Any device that does not exceed the value specified on this parameter is considered a solid state device.

**Syntax:** SGMAXSSDSZ $n$ =*nnnnn*

where  $n$  is a value in the range of 1–8 and is a value less than 32766

**Required:** No

**Default:** SGMAXSSDSZ $n$ =0

**SGP\_DSNINIT=**

**Purpose:** Specifies the size, in number of data sets for the data set name index data space. If the actual number of data sets exceeds 70% of the current size, the number being used is increased by 50%, and the initialization is restarted.

**Syntax:** SGP\_DSNINIT=*nnnnnnnn*

where *nnnnnnnn* is a number in the range 500000-20000000

**Required:** No

**Default:** SGP\_DSNINIT=1000000

**SGP\_EXITBBS=**

**Purpose:** Specifies the number of megabytes to allocate in a scope common data space for the performance exit buffer block.

**Syntax:** SGP\_EXITBBS=*nn*

where  $nn$  is a number in the range 15-99

**Required:** No

**Default:** SGP\_EXITBBS=15

**SGP\_EXITLIB=**

Purpose: Specifies the default library where the performance collector SMF exits reside.

**Note:** EXITLIB in SGPPROC should point to the library that contains the exit load modules: SGPERU83 and SGPERU84. If EXITLIB is *not coded* or is *left as a null* in the started task, it will default to what is coded in SGP\_EXITLIB for which the default is SYS1.LINKLIB. Change SGP\_EXITLIB= to the appropriate load library.

Syntax: SGP\_EXITLIB=xxxxxxx  
where xxxxxx is 1-44 characters

Required: No

Default: SGP\_EXITLIB=SYS1.LINKLIB

**SGP\_MAXCCUS=**

Purpose: Defines the maximum number of control units that are in use during a single collection interval. The minimum value is 1; the maximum value is 310,000.

Syntax: SGP\_MAXCCUS=nnnn

Required: No

Default: SGP\_MAXCCUS=256

**SGP\_MAXDIRS=**

Purpose: Defines the maximum number of directors that are in use during a single collection interval. The minimum value is 1; the maximum value is 20,133,000.

Syntax: SGP\_MAXDIRS=nnnn

Required: No

Default: SGP\_MAXDIRS=256

**SGP\_MAXDSNS=**

**Purpose:** Defines the maximum number of data set names that are in use during a single collection interval. The minimum value is 1; the maximum value is 160,000.

**Syntax:** SGP\_MAXDSNS=*nnnn*

**Required:** No

**Default:** SGP\_MAXDSNS=1000

**SGP\_MAXJOBS=**

**Purpose:** Defines the maximum number of jobs (batch, TSO, and started tasks) that are in use during a single collection interval. The minimum value is 1; the maximum value is 465,000.

**Syntax:** SGP\_MAXJOBS=*nnnn*

**Required:** No

**Default:** SGP\_MAXJOBS=200

**SGP\_MAXLCUS=**

**Purpose:** Defines the maximum number of logical control unit/CHIP combinations in use during an interval. This is the maximum number of actual LCUs in use multiplied by the average number of CHPs carrying data traffic to the LCU. The minimum value is 1; the maximum value is 290,000.

**Note:** If the value is set too low, the system does not set aside enough buffer space to handle all the records. You need to determine a value that provides enough buffer space for LCU records without causing a shortage of dataspace storage for other records.

**Syntax:** SGP\_MAXLCUS=*nnnn*

**Required:** No

**Default:** SGP\_MAXLCUS=256

**SGP\_MAXPOLs=**

**Purpose:** Defines the maximum number of pools that are in use during a single collection interval. The minimum value is 1; the maximum value is 316,000.

**Syntax:** SGP\_MAXPOLs=*nnnn*

Required: No

Default: SGP\_MAXPOLS=256

### **SGP\_MAXPTHS=**

Purpose: Defines the maximum number of CHPIDs that are in use during a single collection interval. The minimum value is 1; the maximum value is 267,000.

Syntax: SGP\_MAXPTHS=nnnn

Required: No

Default: SGP\_MAXPTHS=100

### **SGP\_MAXPVLS=**

Purpose: Defines the maximum number of physical volumes that are in use during a single collection interval. The minimum value is 1; the maximum value is 6,400,000.

Syntax: SGP\_MAXPVLS=nnnn

Required: No

Default: SGP\_MAXPVLS=250

### **SGP\_MAXRRKS=**

Purpose: Defines the maximum number of RAID ranks that are in use during a single collection interval. The minimum value is 1; the maximum value is 512.

Syntax: SGP\_MAXRRKS=nnnn

Required: No

Default: SGP\_MAXRRKS=64

### **SGP\_MAXRSFS=**

Purpose: Defines the maximum number of RVA frames that are in use during a single collection interval. The minimum value is 1; the maximum value is 512.

Syntax: SGP\_MAXRSFS=nnnn

Required: No

Default: SGP\_MAXRSFS=16

**SGP\_MAXSCLS=**

**Purpose:** Defines the maximum number of storage classes that are in use during a single collection interval. The minimum value is 1; the maximum value is 466,000.

**Syntax:** `SGP_MAXSCLS=nnnn`

**Required:** No

**Default:** `SGP_MAXSCLS=256`

**SGP\_MAXVOLS=**

**Purpose:** Defines the total number of online DASD volumes on the OS/390 image being monitored. Note that this is the only `SGP_MAXxxxx` parameter that depends on neither the interval length nor the amount of activity on the system. The minimum value is 1; the maximum value is 438,000.

**Syntax:** `SGP_MAXVOLS=nnnn`

**Required:** No

**Default:** `SGP_MAXVOLS=250`

**SGP\_RDFCOMP=**

**Purpose:** Specifies whether data compression is in effect for records being written to the performance resource data files.

**Syntax:** `SGP_RDFCOMP=Y/N`

**Required:** No

**Default:** `SGP_RDFCOMP=N`

**SGP\_SIBSTK=**

**Purpose:** Identifies the IXFP SIBBATCH parameter member to be used by the MAINVIEW SRM IXFP services for communications with the IXFP address space. The presence of this system parameter value indicates RVA collection is to be activated. This parameter must be specified to activate RVA collection in the performance collector. Without this parameter, the performance collector views will have a physical disk ID and box serial number of question marks.

**Note:** If you have RVA, but do *not* want recording turned on, you must delete the SGP\_SIBSTK variable or make it empty.

**Syntax:** SGP\_SIBSTK=xxxxxxx  
where xxxxxxx is 2-8 characters

**Required:** No

**Default:** None

**SGP\_SMF42=**

**Purpose:** Determines if the SMF 42 record is written to the SMF data set. If set to NO, the historical performance data collector does not allow the record to be written.

**Syntax:** SGP\_SMF42=Y/N

**Required:** No

**Default:** SGP\_SMF42=N

**SGP\_TRACE=**

**Purpose:** Specifies the trace default for the performance collector.

**Syntax:** SGP\_TRACE=xxxxxxx  
where xxxxxxx is one of the following:

FTRACE  
NOTRACE  
NZTRACE  
GTFOUR

**Required:** No



Default: SGP\_TRACE=NOTRACE

**Note:** If no value is specified, NOTRACE is the default. However, GTFOUR is the value that is shipped in the SMMSYSxx startup parameters.

### **SGREADNTVL $n$ =**

Purpose: Specifies the frequency at which the space collector creates a snapshot in core for a specified copy of the space database.

Syntax: SGREADNTVL $n$ = $nnnn$

where  $n$  is a value in the range of 1–8 and where  $nnnn$  is a value in the range 5–9999

Required: No

Default: SG\_READNTVL=30

### **SGRETRYLIM $n$ =**

Purpose: Specifies the number of abend conditions the data collector should ignore for a specified copy of the space database.

Syntax: SGRETRYLIM $n$ = $nnnn$

where  $n$  is a value in the range of 1–8 and where  $nnnn$  is a value from 5–9999

Required: No

Default: SG\_RETRYLIM=10

### **SGSPACHLDR $n$ =**

Purpose: Defines a data set name mask that the space collector can use to identify space holder data sets for a specified copy of the space database.

Syntax: SGSPACHLDR $n$ = $xxxxxxxx$

where  $n$  is a value in the range of 1–8 and where  $xxxxxxxx$  is a space holder data set name mask in the range of 1-44 characters

Required: No

Default: None

**SGSUBTASKS $n$ =**

Purpose: Defines the number of volumes that can be read in parallel for a specified copy of the space collector.

Syntax: SGSUBTASKS $n$ = $nn$

where  $n$  is a value in the range of 1–8 and where  $nn$  is a value in the range 2–10

Required: No

Default: SGSUBTASKS $n$ =3

**SGWRITNTVL $n$ =**

Purpose: Defines the frequency at which snapshots are written to the space database for a specified copy of the space collector.

Syntax: SGWRITNTVL $n$ = $nnnn$

where  $n$  is a value in the range of 1–8 and where  $nn$  is a value in the range 1–1439

Required: No

Default: SG\_WRITNTVL=30

**SIZEISPRIM=**

Purpose: Determines if the SIZE filter/rule list parameter includes only the size of the primary extent or the size of the primary and one secondary extent.

Syntax: SIZEISPRIM=Y/N

Required: No

Default: SIZEISPRIM=Y

**SKIP=**

**Purpose:** Specifies checks to be bypassed during volume switching. There are several conditions in which the SPACVOLA function does not perform a volume switch. Some of these conditions can be bypassed with the SKIP statement. If your installation has an application that can handle data sets that dynamically become multivolume, a SKIP statement can be added to the selection language to bypass requested checks. The NOCHECK operand on the RLST action parameter can also be used to override these checks, and takes precedence over the SKIP statement.

**Note:** Thorough testing and verification that multivolume data sets are usable by the application is recommended before overriding these checks.

**Syntax:** SKIP= (CHECK=(xxxxxx,xxxxxx,...),DD name=xxxxxxx,PROG=xxxxxxx)

where CHECK=(xxxxxx,xxxxxx,...) is one or more of the following options:

EXCP	Bypasses a data set being processed with EXCPs
NOTE	Bypasses a data set being processed with NOTE macros
POINT	Bypasses a data set being processed with POINT macros
DSNAME	Bypasses a data set allocated to another DD statement within the same jobstep
ENQ	Bypasses a permanent data set allocated to a DD statement within another job
DISP	Bypasses a permanent data set being accessed without the use of a catalog
DC	Bypasses a data set that resides on a cached device Under normal conditions, the volume switch will occur only to packs that have the same device characteristics.
CONTIG	Bypasses a data set allocated with a contiguous space requirement

where

DD name=xxxxxxx is any valid file name. If DD name is not specified on the parameter, the file name is not considered in deciding whether to bypass volume switch checks. Only one DD name operand is allowed per SKIP parameter.

PROG=xxxxxxx is any valid program name. If PROG is not specified on the parameter, the program name is not considered in deciding whether to bypass volume switch checks. Only one PROG operand is allowed per SKIP parameter.

**Required:** No

**Default:** None

**SMFID=**

**Purpose:** Specifies the SMF record number for MAINVIEW SRM audit records written to the SMF data set. (Note that SMF message generation is also controlled by the SMF parameter on individual MAINVIEW SRM functions in member SMFUNCxx and by the SMF parameter on filter list specifications.)

**Syntax:** SMFID=*nnn*

where *nnn* is a number in the range 128–255. A zero can be specified to deactivate SMF recording.

**Required:** No

**Default:** none (see explanation that follows)

The sample SMMSYS00 contains a value of 201. If you use the sample as provided, the product will use 201 as the SMFID.

If you delete the SMFID= parameter in SMMSYS00, or build your own SMMSYSxx member with no SMFID= parameter, the product will not write SMF records.

If you change the SMFID= parameter in SMMSYS00 to a different value, or build your own SMMSYSxx member with an SMFID= parameter the product will write SMF records using this value.

**SMS\_ALLOC=**

**Purpose:** Specifies to EasyPOOL that SMSSELECT will be driven during DADSM ALLOCATE. If a POOL is coded in SMSPOOL, the current volume will be compared to the volumes in the POOL. If the current volume is not in a POOL assigned to the data set, the volume will be rejected with a DADSM return code of 4.

**Syntax:** SMS\_ALLOC=Y/N

**Required:** No

**Default:** SMS\_ALLOC=N

**Note:** The new FLST/RLST parameter DADSM\_FUNC should be used to limit the data sets processed by enabling this option.

**SMS\_EXTEND=**

**Purpose:** Specifies to EasyPOOL that SMSSELECT will be driven during DADSM EXTENDNV (new volume). If a POOL is coded in SMSPOOL\_EXT, the current volume will be compared to the volumes in the POOL. If the current volume is not in a POOL assigned to the data set, the volume will be rejected with a DADSM return code of 4.

**Syntax:** SMS\_EXTEND=Y/N

**Required:** No

**Default:** SMS\_EXTEND=N

**Note:** The new FLST/RLST parameter DADSM\_FUNC should be used to limit the data sets processed by enabling this option.

**SMSPool=**

**Purpose:** Specifies the suffix of the SMS pool member. An SMSPOOL<sub>xx</sub> member contains device pool definition parameters. It names SMS subpools and assigns volumes to them.

**Syntax:** SMSPool=<sub>xx</sub>

where <sub>xx</sub> is the two-character suffix of the SMS pool member.

**Required:** No

**Default:** None

**START\_ALL=**

**Purpose:** Specifies whether all components should be started. You can set this keyword to N to prevent all components from starting and to start only SVOS.

**Syntax:** START\_ALL=Y/N

**Required:** No

**Default:** START\_ALL=Y (all licensed products)

**START\_ALLOC=**

Purpose: Specifies whether the Allocation component should be started. You can set this keyword to N to keep the Allocation component from starting in the event of a problem.

Syntax: START\_AUTO=Y/N

Required: No

Default: START\_AUTO=Y (if Allocation component is licensed)

**START\_AUTO=**

Purpose: Specifies whether the automation component should be started. You can set this keyword to N to keep the automation component from starting in the event of a problem.

Syntax: START\_AUTO=Y/N

Required: No

Default: START\_AUTO=Y (if licensed)

**START\_EHSM=**

Purpose: Specifies whether the HSM collector should be started. You can set this keyword to N to keep the HSM collector from starting in the event of a problem.

Syntax: START\_EHSM=Y/N

Required: No

Default: START\_EHSM=Y (if Reporting component is licensed)

**START\_EPOOL=**

Purpose: Specifies whether the EasyPOOL subcomponent should be started. You can set this keyword to N to keep the EasyPOOL subcomponent from starting in the event of a problem.

Syntax: START\_EPOOL=Y/N

Required: No

Default: START\_EPOOL=Y (if Allocation component is licensed)

**START\_ESMS=**

**Purpose:** Specifies whether the EasySMS subcomponent should be started. You can set this keyword to N to keep the EasySMS subcomponent from starting in the event of a problem.

**Syntax:** START\_ESMS=Y/N

**Required:** No

**Default:** START\_ESMS=Y (if Allocation component is licensed)

**START\_RPRT=**

**Purpose:** Specifies whether the Reporting component should be started. You can set this keyword to N to keep the Reporting component from starting in the event of a problem.

**Syntax:** START\_RPRT=Y/N

**Required:** No

**Default:** START\_RPRT=Y (if Reporting component is licensed)

**START\_SGA=**

**Purpose:** Specifies whether the SG-Auto subcomponent should be started. You can set this keyword to N to keep the SG-Auto subcomponent from starting in the event of a problem.

**Syntax:** START\_SGA=Y/N

**Required:** No

**Default:** START\_SGA=Y (if licensed)

**START\_SGC=**

**Purpose:** Specifies whether the application collector should be started. You can set this keyword to N to keep the application collector from starting in the event of a problem.

**Syntax:** START\_SGC=Y/N

**Required:** No

**Default:** START\_SGC=Y (if Reporting component is licensed)

### **START\_SGD=**

**Purpose:** Specifies whether the space collector should be started. You can set this keyword to N to keep the space collector from starting in the event of a problem.

**Syntax:** START\_SGD=Y/N

**Required:** No

**Default:** START\_SGD=Y (if Reporting component is licensed)

### **START\_SGP=**

**Purpose:** Specifies whether the performance collector should be started. You can set this keyword to N to keep the performance collector from starting in the event of a problem.

**Syntax:** START\_SGP=Y/N

**Required:** No

**Default:** START\_SGP=Y (if Reporting component is licensed)

### **START\_X37=**

**Purpose:** Specifies whether StopX37/II subcomponent should be started. You can set this keyword to N to keep the StopX37/II subcomponent from starting in the event of a problem.

**Syntax:** START\_X37=Y/N

**Required:** No

**Default:** START\_X37=Y (if licensed)



**STKSCR=**

Purpose: Specifies the default location of scratch tapes for the STKSUPP function.

Syntax: STKSCR=(xxx,xxx,xxx,xxx)

The four suboperands of STKSCR are

Standard-label tapes

Non-label tapes

ASCII tapes

Non-standard label tapes

For each suboperand, xxx specifies IN (inside a silo), OUT (outside a silo), or a number (specific silo number).

Required: No

Default: None

**STOPX37II=**

Purpose: Specifies whether a full function or limited function of StopX37/II will be started at system start-up.

Syntax: STOPX37II=Y/N

- If STOPX37II is set to N *and* STOPXONLY is set to Y, StopX37/II will run with full functionality.
- If STOPX37II is set to Y, StopX37/II functionality will run with reduced functionality. The following functions will be unavailable:

SPACRLSE

OPTBLKSZ

VIOALLOC

SPACSQTY

SPACCONV

Required: No

Default: STOPX37II=N

**STOPXONLY=**

Purpose:	Specifies whether the StopX37/II will run without the other subcomponents of Allocation.
Syntax:	STOPXONLY=Y/N <ul style="list-style-type: none"><li>• If STOPXONLY is set to N, all components of Allocation will start and run with full functionality.</li><li>• If STOPXONLY is set to Y <i>and</i> STOPX37II is set to N, StopX37/II will run with full functionality without the other subcomponents of Allocation.</li></ul>
Required:	No
Default:	STOPXONLY=N

**SYSLIB=****SYSLIB $n$ =**

Purpose:	<p>Specifies a cataloged <i>data set name for the LPALIB library concatenations that is</i> to be allocated at SVOS startup as a default. This parameter can be overridden by a SYSLIB DD statement in JCL. There is a limit of three data sets that can be concatenated.</p> <p><b>Warning!</b> LPALIB data sets must be the same as they were when the system was last IPLd with a CLPA and/or an MPLA.</p>
Syntax:	<p>SYSLIB=xxxxxxxxxx</p> <p>where xxxxxxxxxxxx is a 1-44 character, fully-qualified cataloged data set name for the LPALIB library concatenations.</p> <p>SYSLIB<math>n</math>=xxxxxxxxxx</p> <p>where <math>n</math> is data set 2 or 3 and where xxxxxxxxxxxx is a 1-44 character, fully-qualified cataloged data set name for the LPALIB library concatenations.</p>
Required:	No
Default:	None

**TAPE\_CA1DSN=**

Purpose: Specifies the data set name of the CA1 data set.

Syntax: TAPE\_CA1DSN=xxxxxxxxxxxx  
where xxxxxxxxxxxx is 1-44 characters

Required: No

Default: None

**TAPE\_CAT=**

Purpose: Specifies the tape management systems available for report generation

Syntax: TAPE\_CAT=(xxxxxxxx,xxxxxxxx,...)

The values are one or more of

- CONTROLT
- CA1
- RMM

That is, TAPE\_CAT=CONTROLT indicates that CONTROL-T is the only system that you are interested in. On the other hand, TAPE\_CAT=(CONTROLT,RMM,CA1) indicates that you have all three systems and you want reports on all three.

Required: No

Default: None

**TAPE\_CCTLTH=**

Purpose: Specifies the high-level qualifier for the CONTROL-T data sets.

Syntax: TAPE\_CCTLTH=xxxxxxxxxxxx  
where xxxxxxxxxxxx is 1-36 characters

Required: No

Default: None

### TAPE\_CHLQ=

Purpose: Specifies the high-level qualifier for the TSCAN data sets.

**Note:** RELEASE cannot be turned on for TSCAN data sets.

Syntax: TAPE\_CHLQ=xxxxxxxxxx  
where xxxxxxxxxxxx is up to 20 characters

Required: No

Default: None

### TAPE\_COLHSM=

Purpose: Specifies whether to collect DFHSM information.

Syntax: TAPE\_COLHSM=Y/N

Required: No

Default: TAPE\_COLHSM=N

### TAPE\_CPRI=

Purpose: Specifies the number of cylinders for the primary allocation.

Syntax: TAPE\_CPRI=nnnn  
where nnnn is 1 to 4369

Required: No

Default: None

### TAPE\_CSEC=

Purpose: Specifies the number of cylinders for the secondary allocation.

Syntax: TAPE\_CSEC=nnnn  
where nnnn is 0 to 4369

Required: No

Default: None

**TAPE\_CTLTRL=**

Purpose: Specifies the release number for Control T.

Syntax: TAPE\_CTLTRL=*x*  
where *x* is a 1 character release number

Required: No

Default: None

**TAPE\_CVOL=**

Purpose: Specifies the volume serial numbers of the volumes to be used for the linear data sets, with a maximum of six volsers.

Syntax: TAPE\_CVOL=(*xxxxxx,xxxxxx,...*)  
where *xxxxxx* is a 1-6 character volume serial number

Required: No

Default: None

**TAPE\_RMMDSN=**

Purpose: Specifies the data set name for the RMM control data set.

Syntax: TAPE\_RMMDSN=*xxxxxxxxxx*  
where *xxxxxxxxxx* is 1-44 characters

Required: No

Default: None

**TAPE\_V\_SUFF=**

**Purpose:** Specifies one or more suffixes of the TAPECAT data set(s) used by the TSCAN image. The specified suffixes prevent the TSCAN from picking up ATL and/or VTS volumes that do not exist on the image it is executing when the TAPECAT data set is shared between images. The format of the TAPECAT data set name is *hlq.VOLCAT.Vx* where *hlq* is the high-level qualifier name and *x* is the suffix.

There is always a TAPECAT name of *hlq.VOLCAT.VGENERAL*. This TAPECAT data set will always be read, and it may contain volume serials. It also contains the names of the libraries that the volume serials are in.

**Syntax:** TAPE\_V\_SUFF=xxxxxxxxxxxxxx

where xxxxxxxxxxxxxxxx is 1-36 characters – 26 letters and digits 0 through 9.

The following is an example of this parameter when the TAPECAT data sets to be read are *hlq.VOLCAT.VA* and *hlq.VOLCAT.VS*:

TAPE\_V\_SUFF=AS

The TAPECAT data set of *hlq.VOLCAT.VGENERAL* will *always* be read. If this parameter is not specified, the TSCAN process will read all of the TAPECAT data sets.

**Required:** No

**Default:** None

**TAPEGENR=**

**Purpose:** Specifies tape device generic names that some EasyPOOL functions will intercept.

If you want to intercept all tape requests, specify ALLTAPE as the first generic name. (However, you cannot use ALLTAPE when JCLEXT=N.) TAPEGENR affects functions DSNCHECK and SETEXPDT.

**Syntax:** TAPEGENR=(xxxxxxxx,xxxxxxxx,xxxxxxxx,...)

where xxxxxxxx is a tape device name in 1-8 characters

**Required:** No

Default: None

**Note:** If this parameter is not coded, all data sets with tape generics or esoterics will be processed. If any parameter is coded for TAPEGENR, only those tape unit names in TAPEGENR will be processed, so all generic/esoteric unit names that are to be processed should be specified. PROCOLD determines whether EasyPOOL intercepts DD parameters associated with existing data sets. Specify PROCOLD=Y if you want to convert unit information for existing tape data sets. EasyPOOL can then intercept DD parameters for existing data sets that also specify UNIT.

### **TRACEDD=**

**Purpose:** Traces all MAINVIEW SRM functions for the jobstep containing the specified DD name. This is the same type of filter/rule list trace as produced by the TRACE parameter for the SMFUNCxx function definition; however, using TRACEDD, *all* MAINVIEW SRM functions will be traced for a single jobstep, based on the presence of a JCL DD name.

**Syntax:** TRACEDD=xxxxxxxx  
where xxxxxxxx is a 1–8 character DD name.

**Required:** No

**Default:** None

### **TRKCYL=**

**Purpose:** Specifies the number of tracks per cylinder for the default device type. The value specified for 3380/3390/9345 devices should be 15. (Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.)

TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.

**Syntax:** TRKCYL=nnnnn  
where nnnnn is a number in the range of 1-99999

**Required:** Yes

**Default:** None

**TRKLEN=**

**Purpose:** Specifies the number of bytes per track for the default device type. Valid values are:

3380 - 47,476

3390 - 56,664

9345 - 46,456

**Note:** Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.

TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.

**Syntax:** TRKLEN=nnnnnnnn

where *nnnnnnnn* is a 1 to 7 digit number.

**Required:** Yes

**Default:** None

**USECAT=**

**Purpose:** Specifies whether the catalog name is used as a selection criteria in any MAINVIEW SRM ACS replacement function (SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG). Can cause an embrace with catalog functions.

**Syntax:** USECAT=Y/N

**Required:** No

**Default:** USECAT=N

**USEMVI=**

**Purpose:** Supports users who install and run only the StopX37/II Stand-Alone product. If you specify USEMVI=YES or Y, then you must specify the BBI3\_SSID parameter. If you specify USEMVI=NO or N, then BBI3\_SSID does not have to be specified; if it is specified, MAINVIEW SRM ignores it and does not attempt to attach to the CAS.

**Syntax:** USEMVI=Y/N or *Yes/No*



Required: No

Default: USEMVI=Yes

### **VAR=**

**Purpose:** Specifies the suffix of the SMVARS<sub>xx</sub> member. SMVARS<sub>xx</sub> contains variables definition parameters. The values of defined variables are substituted in MAINVIEW SRM selection statements to simplify the specification of large selection criteria used in multiple statements.

**Syntax:** VAR=<sub>xx</sub>

where <sub>xx</sub> is any two-character string. A single character is not allowed.

Required: No

Default: None

### **VSAM\_ADJ4GB=**

**Purpose:** Controls StopX37/II recovery for non-extended VSAM data sets when new extent will exceed the 4GB limit. VSAM\_ADJ4GB=YES allows StopX37/II to adjust the requested allocation amount to fit within the 4GB maximum limit.

**Syntax:** VSAM\_ADJ4GB=YES/NO

Required: No

Default: VSAM\_ADJ4GB=YES

### **VSAMJCL=**

**Purpose:** Controls the level of processing of VSAM data sets by EasyPOOL.

With VSAMJCL=CLUSTER, the EasyPOOL functions are invoked for the VSAM cluster if the volume list is defined at the cluster level, or if volume lists of the two components are identical to each other. The EasyPOOL functions are invoked at the component level if the volume list is defined at the component level and are *not* identical to each other.

VSAMJCL=COMPONENT causes the JCL function to process at the component level regardless of how the volume list is defined.

**Syntax:** VSAMJCL=CLUS/COMP

Required: No

Default: VSAMJCL=COMP

### **VSAMLIMWARN=**

Purpose: Specifies the percentage value to be used before issuing the MAINVIEW SRM 4GB limit message. This is an informational message to show how close a non-extended format VSAM file is to the 4GB limit.

Syntax: VSAMLIMWARN=*xx*  
where *xx* is a two-digit number in the range 0–99

Required: No

Default: VSAMLIMWARN=90

### **VSAMPRIM=**

Purpose: Specifies that volume additions to a VSAM file (by SPACVOLA) will use the primary allocation size instead of the secondary.

Syntax: VSAMPRIM=Y  
Use the *primary* allocation size.  
  
VSAMPRIM=N  
Use the *secondary* allocation size.

Required: No

Default: VSAMPRIM=N

### **VSAMZSEC=**

Purpose: Specifies which StopX37/II function controls recoveries for VSAM out-of-space conditions when no secondary allocation amount was specified when the data set was defined. Specifying VSAMZSEC=Y indicates that the SPACSECA function controls whether recovery is allowed. VSAMZSEC=N specifies that the SPACVOLA function will determine whether recovery is allowed. If an out-of-space condition occurs for a VSAM data set because no secondary allocation amount was specified, and VSAMZSEC=Y is specified, StopX37/II will only recover from the error if the SPACSECA function is active for the same data set. Specifying VSAMZSEC=Y and not activating the SPACSECA function for a data set indicates that VSAM data sets that do not have a secondary allocation amount cannot be recovered.

Syntax: VSAMZSEC=Y/N

Required: No

Default: VSAMZSEC=Y

**VSCAN\_AGER1=**

Purpose: Specifies the high end of age range one. Data sets that have not been used between zero days and this value fall into AGER1. The value is optional and is a 1–3 digit number. The number must be less than the remaining ranges.

Syntax: VSCAN\_AGER1=*nnn*  
where *nnn* is 1 to 999

Required: No

Default: None

**VSCAN\_AGER2=**

Purpose: Specifies the high end of age range two. Data sets that have not been used between AGER1 and this value fall into AGER2. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

Syntax: VSCAN\_AGER2=*nnn*  
where *nnn* is 0 to 999

Required: No

Default: None

**VSCAN\_AGER3=**

Purpose: Specifies the high end of age range three. Data sets that have not been used between AGER2 and this value fall into AGER3. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

Syntax: VSCAN\_AGER3=*nnn*  
where *nnn* is 0 to 999

Required: No

Default: None

**VSCAN\_AGER4=**

**Purpose:** Specifies the high end of age range four. Data sets that have not been used between AGER3 and this value fall into AGER4. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

**Syntax:** VSCAN\_AGER4=*nnn*  
where *nnn* is 0 to 999

**Required:** No

**Default:** None

**VSCAN\_AGER5=**

**Purpose:** Specifies the high end of age range five. Data sets that have not been used between AGER4 and this value fall into AGER5. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

**Syntax:** VSCAN\_AGER5=*nnn*  
where *nnn* is 0 to 999

**Required:** No

**Default:** None

**VSCAN\_AGER6=**

**Purpose:** Specifies the high end of age range six. Data sets that have not been used between AGER5 and this value fall into AGER6. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

**Syntax:** VSCAN\_AGER6=*nnn*  
where *nnn* is 0 to 999

**Required:** No

**Default:** None

**VSCAN\_AGER7=**

**Purpose:** Specifies the high end of age range seven. Data sets that have not been used between AGER6 and this value fall into AGER7. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

**Syntax:** VSCAN\_AGER1=*nnn*  
where *nnn* is 0 to 999

**Required:** No

**Default:** None

**VSCAN\_AGER8=**

**Purpose:** Specifies the high end of age range eight. Data sets that have not been used between AGER7 and this value fall into AGER8. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

**Syntax:** VSCAN\_AGER8=*nnn*  
where *nnn* is 0 to 999

**Required:** No

**Default:** None

**VSCAN\_AGER9=**

**Purpose:** Specifies the high end of age range nine. Data sets that have not been used between AGER8 and this value fall into AGER9. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding ranges and less than the remaining ranges. When set to zero, the range is ignored. However, all remaining ranges must also be set to zero.

**Note:** All data sets that exceed range nine fall into the tenth range. The tenth range is automatically set; it cannot be set by a user.

**Syntax:** VSCAN\_AGER9=*nnn*  
where *nnn* is 0 to 999

**Required:** No

Default: None

**VSCAN\_PCTR1=**

Purpose: Percent used refers to how full the data set is. Specifies the high end of percentage used range one. Data sets that are between zero and this value fall into PCTR1. The value is optional and is a 1–3 digit number. The value must be less than the remaining percentage used values.

Syntax: VSCAN\_PCTR1=*nnn*  
where *nnn* is 1 to 100

Required: No

Default: None

**VSCAN\_PCTR2=**

Purpose: Specifies the high end of percentage used range two. Data sets that are between PCTR1 and this value fall into PCTR2. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

Syntax: VSCAN\_PCTR2=*nnn*  
where *nnn* is 0 to 100

Required: No

Default: None

**VSCAN\_PCTR3=**

Purpose: Specifies the high end of percentage used range three. Data sets that are between PCTR2 and this value fall into PCTR3. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

Syntax: VSCAN\_PCTR3=*nnn*  
where *nnn* is 0 to 100

Required: No

Default: None

**VSCAN\_PCTR4=**

**Purpose:** Specifies the high end of percentage used range four. Data sets that are between PCTR3 and this value fall into PCTR4. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

**Syntax:** VSCAN\_PCTR4=*nnn*  
  
where *nnn* is 0 to 100

**Required:** No

**Default:** None

**VSCAN\_PCTR5=**

**Purpose:** Specifies the high end of percentage used range five. Data sets that are between PCTR4 and this value fall into PCTR5. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

**Syntax:** VSCAN\_PCTR5=*nnn*  
  
where *nnn* is 0 to 100

**Required:** No

**Default:** None

**VSCAN\_PCTR6=**

**Purpose:** Specifies the high end of percentage used range six. Data sets that are between PCTR5 and this value fall into PCTR6. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

**Syntax:** VSCAN\_PCTR6=  
  
where *nnn* is 0 to 100

**Required:** No

**Default:** None

**VSCAN\_PCTR7=**

**Purpose:** Specifies the high end of percentage used range seven. Data sets that are between PCTR6 and this value fall into PCTR7. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

**Syntax:** VSCAN\_PCTR7=*nnn*  
where *nnn* is 0 to 100

**Required:** No

**Default:** None

**VSCAN\_PCTR8=**

**Purpose:** Specifies the high end of percentage used range eight. Data sets that are between PCTR7 and this value fall into PCTR8. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

**Syntax:** VSCAN\_PCTR8=*nnn*  
where *nnn* is 0 to 100

**Required:** No

**Default:** None

**VSCAN\_PCTR9=**

**Purpose:** Specifies the high end of percentage used range nine. Data sets that are between PCTR8 and this value fall into PCTR9. The value is optional and is a 0–3 digit number. The number if specified, must be higher than the preceding values and less than the remaining values. When set to zero, the value is ignored. However, all remaining values must also be set to zero.

**Note:** All data sets that exceed range nine fall into the tenth range. The tenth range is automatically set; it cannot be set by a user.

**Syntax:** VSCAN\_PCTR9=*nnn*  
where *nnn* is 0 to 100

**Required:** No



Default: None

**VSCAN\_SIZR1=**

Purpose: Specifies the high end of size range one. Data sets that are between zero and this value fall into SIZR1. The value is optional and is a 1–9 digit number (in kilobytes). The value must be less than the remaining sizes.

Syntax: VSCAN\_SIZR1=nnnnnnnnnn

where *nnn* is 1 to 999999999

Required: No

Default: None

**VSCAN\_SIZR2=**

Purpose: Specifies the high end of size range two. Data sets that are between SIZR1 and this value fall into SIZR2. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

Syntax: VSCAN\_SIZR2=nnnnnnnnnn

where *nnn* is 0 to 999999999

Required: No

Default: None

**VSCAN\_SIZR3=**

Purpose: Specifies the high end of size range three. Data sets that are between SIZR2 and this value fall into SIZR3. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

Syntax: VSCAN\_SIZR3=nnnnnnnnnn

where *nnn* is 0 to 999999999

Required: No

Default: None

**VSCAN\_SIZR4=**

**Purpose:** Specifies the high end of size range four. Data sets that are between SIZR3 and this value fall into SIZR4. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

**Syntax:** VSCAN\_SIZR4=*nnnnnnnnnn*  
where *nnn* is 0 to 999999999

**Required:** No

**Default:** None

**VSCAN\_SIZR5=**

**Purpose:** Specifies the high end of size range five. Data sets that are between SIZR5 and this value fall into SIZR6. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

**Syntax:** VSCAN\_SIZR5=*nnnnnnnnnn*  
where *nnn* is 0 to 999999999

**Required:** No

**Default:** None

**VSCAN\_SIZR6=**

**Purpose:** Specifies the high end of size range six. Data sets that are between SIZR5 and this value fall into SIZR6. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

**Syntax:** VSCAN\_SIZR6=*nnnnnnnnnn*  
where *nnn* is 0 to 999999999

**Required:** No

**Default:** None

**VSCAN\_SIZR7=**

**Purpose:** Specifies the high end of size range seven. Data sets that are between SIZR6 and this value fall into SIZR7. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

**Syntax:** VSCAN\_SIZR7=nnnnnnnnnn  
where *nnn* is 0 to 999999999

**Required:** No

**Default:** None

**VSCAN\_SIZR8=**

**Purpose:** Specifies the high end of size range eight. Data sets that are between SIZR7 and this value fall into SIZR8. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

**Syntax:** VSCAN\_SIZR8=nnnnnnnnnn  
where *nnn* is 0 to 999999999

**Required:** No

**Default:** None

**VSCAN\_SIZR9=**

**Purpose:** Specifies the high end of size range nine. Data sets that are between SIZR8 and this value fall into SIZR9. The value is optional and is a 0–9 digit number (in kilobytes). The number if specified, must be higher than the preceding sizes and less than the remaining sizes. When set to zero, the size is ignored. However, all remaining sizes must also be set to zero.

**Note:** All data sets that exceed range nine fall into the tenth range. The tenth range is automatically set; it cannot be set by a user.

**Syntax:** VSCAN\_SIZR9=nnnnnnnnnn  
where *nnn* is 0 to 999999999

**Required:** No

Default: None

**VSCAN\_MNTSK=**

Purpose: Specifies the minimum number of tasks (TCBs) used by the VTOC scan to perform the collection.

Syntax: VSCAN\_MNTSK=*nn*  
where *nn* is 2 to 30

Required: No

Default: VSCAN\_MNTSK=2

**VSCAN\_MXTSK=**

Purpose: Controls the number of tasks (TCBs) involved in scanning VTOCs for VTOC reporting and automation requests. It can also be used to affect the response given to automation requests.

Syntax: VSCAN\_MXTSK=*nn*  
where *nn* is 2 to 30

Required: No

Default: VSCAN\_MXTSK=8

**VSCAN\_OINDX=**

Purpose: Specifies the prefix name of the VTOC scan collection data set.  
*Dyymmdd.Thhmmss* is appended to the prefix to complete the full data set name.

Syntax: VSCAN\_OINDX=xxxxxxxxxxxx...  
where xxxxxxxxxxxx... is 1 to 19 characters, following standard data set naming conventions

Required: Yes

Default: None

**VSCAN\_OPRI=**

**Purpose:** Specifies the primary allocation size in cylinders for the VTOC scan collection data set.

**Syntax:** VSCAN\_OPRI=nnnn  
where nnnn is 1 to 4369

**Required:** No

**Default:** VSCAN\_OPRI=10

**VSCAN\_OSEC=**

**Purpose:** Specifies the secondary allocation size in cylinders for the VTOC scan collection data set.

**Syntax:** VSCAN\_OSEC=nnnn  
where nnnn is 1 to 4369

**Required:** No

**Default:** VSCAN\_OSEC=10

**VSCAN\_OUNIT=**

**Purpose:** Specifies the device type of the VTOC scan collection data set. The VSCAN\_OUNIT parameter must specify a direct access device type. If it does not, when a VTOC scan is initiated, message SVO5009E is issued and the scan is terminated.

**Syntax:** VSCAN\_OUNIT=xxxxxxxx  
where xxxxxxxx is a 1- to 8-character valid direct access device number or name defined in your environment

**Required:** Yes

**Default:** None

**VSCAN\_OVOL=**

**Purpose:** Specifies the volume serial number of the VTOC scan collection data set.

**Syntax:** VSCAN\_OVOL=xxxxxx

where *xxxxxx* is a 1- to 6- character valid volume serial number defined in your environment

Required: No

Default: None

#### **VSCAN\_TPRI=**

Purpose: Specifies the primary allocation size in cylinders for the VTOC scan temporary data set.

Syntax: **VSCAN\_TPRI=nnnn**

where *nnnn* is 1 to 4369

Required: No

Default: **VSCAN\_TPRI=10**

#### **VSCAN\_TSEC=**

Purpose: Specifies the set secondary allocation size in cylinders for the VTOC scan temporary data.

Syntax: **VSCAN\_TSEC=nnnn**

where *nnnn* is 1 to 4369

Required: No

Default: **VSCAN\_TPRI=10**

#### **VSCAN\_TUNIT=**

Purpose: Specifies the device type for the VTOC scan temporary data set.

Syntax: **VSCAN\_TUNIT=xxxxxxxx**

where *xxxxxxxx* is a 1- to 8-character valid device number or name defined in your environment

Required: Yes

Default: None

**VSCAN\_TVOL=**

Purpose: Specifies the volume serial number for the VTOC scan temporary data set.

Syntax: VSCAN\_TVOL=xxxxxx

where xxxxxx is a 1- to 6-character valid volume serial number defined in your environment

Required: No

Default: None

**WTODC=**

Purpose: Specifies the message descriptor codes to be assigned to messages written by MAINVIEW SRM. Examine the DESC keyword parameter on the WTO statement found in *MVS Supervisor Services and Macro Instructions* for an explanation of description codes.

Syntax: See WTO macro in *MVS Supervisor Services and Macro Instructions*. The parameter accepts 1-16 characters.

Required: No

Default: None

**WTORC=**

Purpose: Specifies the routing code to be assigned to the message text. For more information, see the WTO macro's ROUTCDE= parameter in the *MVS Supervisor Services and Macro Instructions*.

Syntax: WTORC=nn

where nn is a number from 0 to 16. If you need more than one code, enclose them in parentheses, separated with commas.

Required: No

Default: WTORC=0

**X37POOL=**

Purpose: Specifies which volume will be used by StopX37/II to determine the POOL name in EOVS processing.

Syntax: X37POOL=NEW/ORIG

Required: No

Default: X37POOL=ORIG

**X37RLS=**

Purpose: Specifies whether StopX37/II end-of-volume functions are to be activated for VSAM Record-Level-Sharing (RLS) data sets.

StopX37/II functions for RLS data sets will work only in a CICS environment. Refer to additional installation requirements for RLS data sets under in the *MAINVIEW SRM StopX37/II User Guide and Reference*.

Syntax: X37RLS=YES/NO

YES = Activate StopX37/II for RLS data sets.

NO = No StopX37/II processing for RLS data sets.

Required: No

Default: X37RLS=NO



# Pool Member Parameters

**SMPOOLxx** SMPOOLxx organizes DASD volumes into pools. For more information, see Chapter 6, “Defining a Pool.”

## Parameter Quick Reference

The following tables provide a brief description of SET statements used in SMPOOLxx and a brief description of INC and EXC statements used in SMPOOLxx. Detailed descriptions of each parameter are listed in alphabetical order after the tables.

**Table D-2 SET Statement Pool Parameter Quick Reference**

Parameter	Required	Description
POOLNAME=xxxxxxx	Yes	Name to be assigned to pool
USELIMIT=nnn	No	Upper space threshold for new allocations
SGDCollect=YES NO	No	Specifies whether a pool is processed by the space collector
SGDCollectn=YES NO	No	Specifies whether a pool is processed by an alternate space collector indicated by a suffix of <i>n</i> .
TYPE=xxxx	No	Device type

**Table D-3 INC and EXC Statement Pool Parameter Quick Reference**

Parameter	Required	Description
ADR=xxxx	No	Device address of tape unit in pool
ADR=(xxxx,xxxx,...)	No	Multiple tape device addresses (up to 15)
VOL=xxxxxx	No	Volume serial number of device in pool
VOL=(xxxxxx,xxxxxx,...)	No	Multiple volume serial numbers (up to 15)

## Parameter Explanations

### ADR=

**Purpose:** Specifies the device addresses of tape units to be included in or excluded from the pool. MAINVIEW SRM name masking can be used.

**Syntax:** ADR=xxxx or ADR=(xxxx,xxxx,...)

where xxxx is a 4-byte character string. Up to 15 addresses can be specified by enclosing the numbers in parentheses.

Four-character device addresses were introduced with MVS/ESA 5.1. You must specify a full four-character address even if you are running an earlier release of MVS.

**Required:** No

**Default:** None

### POOLNAME=

**Purpose:** Specifies the name of the pool. (Note that this definition is independent of the MVSCP.) The pool names specified need not be defined to MVS as esoteric device names.

**Syntax:** POOLNAME=xxxxxxxx

where xxxxxxxx is a 1–8 character string.

In addition to the 1–8 character string, the following can be specified to derive the pool name dynamically from the device being processed:

POOLNAME=&xxxxxxxx/(start,end)

where &xxxxxxxx is one of the following:

&VOL  
&UNIT  
&MNTYPE  
&STOGROUP  
&STORGRP

Start and end are used to specify which characters will be used in the pool name. If start and end are not used, all characters will be used. For example, if the volume serial number is ABC123 and POOLNAME=&VOL is specified, the pool name will be ABC123. If POOLNAME=&VOL(1,3) is specified, the pool name will be ABC. If POOLNAME=&VOL(3,6) is specified, the pool name will be C123.

The &VOL, &UNIT, and &MNTYPE variables are supported only by the space collector. When variable-named pools are used, the default for SGDCOLLECT is YES. If NO is specified on SGDCOLLECT, it is ignored. These pools appear only under the Space Collector Pools option of the Space Utilization menu.

The &STOGROUP and &STORGRP variables are supported throughout MAINVIEW SRM.

Required: Yes

Default: None

#### **SGDCOLLECT=**

Purpose: Specifies whether a pool is processed by the space collector.

Syntax: SGDCOLLECT=Y/N

Required: No

Default: No

#### **SGDCOLLECT<sub>n</sub>=**

Purpose: Specifies whether a pool is processed by an alternate space collector. The alternate data collector to be used is identified by the suffix *n*.

Syntax: SGDCOLLECT<sub>n</sub>=Y/N

Required: No

Default: No

#### **TYPE=**

Purpose: Specifies the type of device.

Syntax: TYPE=xxxxx

where xxxxx is one of the following values: DASD, 3420, 3480, 3490.

Types 3420/3480/3490 are used to define tape pools for the TAPEPOOL function.

Required: No

Default: TYPE=DASD

### **USELIMIT=**

**Purpose:** Specifies an upper space limit for DASD volumes in a pool. MAINVIEW SRM attempts to prevent allocation of a new data set to a given DASD volume if that allocation would cause the volume USELIMIT threshold to be exceeded. This threshold is provided to ensure sufficient space on a volume for existing data sets to be extended with secondary extents. The USELIMIT parameter is similar to the high allocation threshold provided by DFSMS.

This parameter does not apply to tape devices or to DFSMS-managed DASD volumes.

The USELIMIT parameter on a pool will not prevent a pool assignment, even if a volume within the USELIMIT percentage cannot be found. In this case, the last volume found that would satisfy the primary allocation will be selected.

Note that USELIMIT applies only to primary allocation processing; during allocation of secondary extents, the USELIMIT is not enforced. If primary allocations are consistently too small for all data sets on a volume, thus requiring extensive secondary allocations, it is still possible to exceed the USELIMIT and fill a volume.

USELIMIT will only be applied by DASDPOOL when the VOLSEL parameter has been specified. USELIMIT is also applied by the SPACVOLA function.

**Syntax:** USELIMIT=*nnn*

where *nnn* is a number in the range 1–100.

Required: No

Default: None

### **VOL=**

**Purpose:** Specifies the volume serial numbers of DASD devices to be included in or excluded from the pool. MAINVIEW SRM name masking can be used.

**Syntax:** VOL=*xxxxxx* or VOL=(*xxxxxx,xxxxxx,...*)

where *xxxxxx* is a 1–6 character string. Up to 15 volumes can be specified by enclosing the numbers in parentheses.

Required: No

Default: None

## SMS Subpool Member Parameters

**SMSPOLxx** Organizes SMS-managed DASD volumes into subpools; SMS subpools are only used by EasyPOOL. For more information, see the *MAINVIEW SRM EasyPOOL User Guide and Reference*.

### Parameter Quick Reference

The following tables provide a brief description of SET statements used in SMSPOLxx and a brief description of INC and EXC statements used in SMSPOLxx. Detailed descriptions of each parameter are listed in alphabetical order after the tables.

**Table D-4 SET Statement SMS Pool Parameters**

Parameter	Required	Description
POOLNAME=xxxxxxx	Yes	Name to be assigned to SMS subpool

**Table D-5 INC and EXC Statement SMS Pool Parameters**

Parameter	Required	Description
VOL=xxxxxx	No	Volume serial number of device in subpool
VOL=(xxxxxx,xxxxxx,...)	No	Multiple volume serial numbers (up to 15)

### Parameter Explanations

#### **POOLNAME=**

**Purpose:** Specifies the name of the SMS subpool. (Note that this definition is independent of the MVSCP.) The SMS subpool names specified need not be defined to MVS as esoteric device names.

Syntax: POOLNAME=xxxxxxx  
where xxxxxxx is a 1–8 character string.

Required: Yes

Default: None

#### **VOL=**

Purpose: Specifies the volume serial numbers of SMS-managed DASD devices to be included in or excluded from the subpool. MAINVIEW SRM name masking can be used.

Syntax: VOL=xxxxxx or VOL=(xxxxxx,xxxxxx,...)  
where xxxxxx is a 1–6 character string. Up to 15 volumes can be specified by enclosing the numbers in parentheses.

Required: No

Default: None

## **Calendar Member Parameters**

**SMCALSxx** SMCALSxx defines non-working days for DFHSM migration processing and other date-related processing. For more information, see Chapter 8, “Defining a Calendar.”

## **Parameter Quick Reference**

The following tables provide a brief description of SET statements used in SMCALSxx and a brief description of INC and EXC statements used in SMCALSxx. Detailed descriptions of each parameter are listed in alphabetical order after the tables.

**Table D-6 SET Statement Calendar Parameters**

Parameter	Required	Description
YEAR=nnnn	Yes	Year to which the following days apply

**Table D-7 INC and EXC Statement Calendar Parameters**

Parameter	Required	Description
FREE= <i>nn.nn-nn.nn</i>	No	From-to range of non-working (free) days
MON= <i>F/W</i>	No	Day of week
TUE= <i>F/W</i>	No	Day of week
WED= <i>F/W</i>	No	Day of week
THU= <i>F/W</i>	No	Day of week
FRI= <i>F/W</i>	No	Day of week
SAT= <i>F/W</i>	No	Day of week
SUN= <i>F/W</i>	No	Day of week

## Parameter Explanations

### **YEAR=**

Purpose: Specifies the year being defined.

Syntax: YEAR=*nnnn*

where *nnnn* is a four-digit year in the range 1900–2100.

Required: Yes

Default: None

### **FREE=**

Purpose: Specifies a single date or a date range that represents non-working days (days that are not considered as usage days by DFHSM).

Note that the DATEFMT parameter in SMMSYSxx does not apply to date specifications in SMCALSxx.

Syntax: FREE=*nn.nn-nn.nn*

where *nn.nn* is a date specification of the form dd.mm, where dd and mm both are two-digit numbers. For example:

07.12	December 7
15.02	February 15
01.07-05.07	July 1–5

Required: No

Default: None

### **MON-SUN=**

Purpose: Specifies that a specific day of the week is either a non-working (free) day or a working (usage) day.

Syntax: MON=F/W

where F identifies a non-working day, and W identifies a working day.

Required: No

Default: None

## **Variable Member Parameters**

**SMVARSxx** SMVARSxx defines variables to contain MAINVIEW SRM selection parameters. These variables can be included in filter and rules lists. For more information, see Chapter 7, “Using Variables.”

## **Parameter Quick Reference**

The following tables provide a brief description of SET statements used in SMVARSxx and a brief description of INC and EXC statements used in SMVARSxx. Detailed descriptions of each parameter are listed in alphabetical order after the tables.

**Table D-8 SET Statement Variable Parameters**

Parameter	Required	Description
VARIABLE=xxxxxxx	Yes	Name assigned to variable

**Table D-9 INC and EXC Statement Variable Parameters**

Parameter	Required	Description
VALUE=xxxxxxxxxxxx	Yes	Any values valid in selection parameters

SMVARSxx is an optional member.



## Parameter Explanations

### **VARIABLE=**

Purpose:	Specifies the name of the variable.
Syntax:	<b>VARIABLE=</b> <i>xxxxxxxx</i>  where <i>xxxxxxxx</i> is a 1- to 30-character string.
Required:	Yes
Default:	None

### **VALUE=**

Purpose:	Specifies a value for the variable.
Syntax:	<b>VALUE=</b> <i>xxxxxxxxxxxxxxxxxx</i>  where <i>xxxxxxxxxxxxxxxxxx</i> is any character string, with no embedded blanks.
Required:	Yes—at least one value must be declared for a variable.
Default:	None

## Function Member Parameters

<b>SMFUNCxx</b>	SMFUNCxx defines and activates functions. A function must have an entry in SMFUNCxx to be available to the executing MAINVIEW SRM subsystem. A function's parameters include specification of a filter list member and a rule list member (if required). These two PARMLIB members give tremendous flexibility in applying a function's processing to data resources. For more information, see Chapter 4, “Defining the SMFUNCxx Member and Functions.”
-----------------	--

## Subordinate Members

SMFLST $xx$ , SMRLST $xx$

## Parameter Quick Reference

The following table provides a brief description of SET statements used in SMVARS $xx$ . Detailed descriptions of each parameter are listed in alphabetical order after the tables.

**Table D-10 SET Statement Function Parameters**

Parameter	Required	Description
NAME=xxxxxxxx	Yes	MAINVIEW SRM-assigned name of the function
FLST= $xx$	No	Suffix of filter list member SMFLST $xx$
RLST= $xx$	No	Suffix of rule list member SMRLST $xx$
ACTIVE=Y/N	Yes	Status of the function
MSG=I/W/E/S/N	Yes	Level of messages to be generated
SMF=I/W/E/S/N	Yes	Level of messages to be written to SMF
TRACE=xxxxxxxx	No	Jobname of traced MAINVIEW SRM actions
DESC='xxxxxxxxxx xxxxxx'	No	Description of function

SMFUNC $xx$  is required. MAINVIEW SRM does nothing without defined function parameters.

## Parameter Explanations

### NAME=

**Purpose:** Specifies the name of the function. Function names are assigned within MAINVIEW SRM code.

**Syntax:** NAME=xxxxxxxx  
  
where xxxxxxxx is a 1–8 character string of a MAINVIEW SRM function assigned by BMC Software.

**Required:** Yes

**Default:** None

**FLST=**

**Purpose:** Specifies the suffix of the filter list PARMLIB member (SMFLST $xx$ ) for this function. The filter list allows selection of resources that are affected by the function. If no filter list member is specified, no resources are selected for the function.

**Syntax:** FLST= $xx$

where  $xx$  is any two-character string. A single character is not allowed.

**Required:** No

**Default:** None

**RLST=**

**Purpose:** Specifies the suffix of the rule list PARMLIB member (SMRLST $xx$ ) for this function. The rule list allows specification of how the function is applied to selected resources. If no rule list is specified, the function default processing is applied to all resources selected by the filter list parameters. However, if there is no default processing by the function (that is, an action parameter is required for the function to have effect), a rule list must be specified (SET and INC parameters) for any processing to take place.

Note that the following functions do not use a rule list: FORCECAT, MODDELET, OPENEMPT, SMSMCREN, SUPJSCAT, SUPVOLRF, and TAPEDEFR.

**Syntax:** RLST= $xx$

where  $xx$  is any two-character string. A single character is not allowed.

**Required:** Yes, if the function uses a rule list; otherwise, no.

**Default:** None

**ACTIVE=**

**Purpose:** Specifies the status of the function. If ACTIVE=NO is specified, the function has no effect, regardless of any specifications in the filter or rule list members. ACTIVE=YES must be specified for the function to provide any MAINVIEW SRM services.

**Syntax:** ACTIVE=Y/N

**Required:** Yes

Default: None

**MSG=**

**Purpose:** Specifies the default message generation option for the function. Information and error messages can be produced, or all messages can be suppressed. Note that the MSG option on the filter list SET command overrides this option for selected resources.

**Syntax:** MSG=I/W/E/S/N

where

*I* = Information and error messages

*W* = Warning messages

*E* = Error messages only

*S* = Severe error messages

*N* = No messages

**Required:** Yes

**Default:** None

**SMF=**

**Purpose:** Specifies the SMF message generation option for the function. Information and error messages can be written to the SMF data set, or all messages can be omitted from the SMF data set. Note that the SMF option on the filter list SET command overrides this option for selected resources.

**Syntax:** SMF=I/W/E/S/N

where

*I* = Information and error messages

*W* = Warning messages

*E* = Error messages only

*S* = Severe error messages

*N* = No messages

**Required:** Yes

**Default:** None

**TRACE=**

**Purpose:** Specifies that, for the identified job, all filter and rule list processing for the function is to be traced by writing MAINVIEW SRM messages.

**Syntax:** TRACE=xxxxxxx

where xxxxxxx is a 1–8 character jobname (including TSO session ID or started task name). Note that the name of the job to be traced must match this parameter value exactly; *name masking does not apply to this parameter*.

**Required:** No

**Default:** None

**DESC=**

**Purpose:** Specifies a short description for the function. This description appears in the ISPF panels where the function is displayed.

**Syntax:** DESC='xxxxxxx'

where xxxxxxx is a quoted string up to 46 characters long.

**Required:** Yes

**Default:** None

# Diagnostic Member Parameters

## SMDIAGxx

SMDIAGxx aids in diagnosing problems in MAINVIEW SRM modules. For more information, see Chapter 9, “Using Diagnostics.”

## Parameter Quick Reference

The following tables provide a brief description of SET statements used in SMDIAGxx and brief description of INC and EXC statements used in SMDIAGxx. Because this member is used only at the direction of BMC Software Customer Support, the parameters are not described in detail.

**Table D-11 SET Statement Diagnostic Parameters**

Parameter	Required	Description
ABEND=Y/N	No	forces S0C3 abend when a particular module is entered
DEBUG=Y/N	No	optional debugging information from a MAINVIEW SRM module
DUMP=Y/N	No	issues SDUMP if program abend occurs
IGNORE=Y/N	No	skips this function
MODTRC=Y/N	No	MAINVIEW SRM module trace
TRACE=Y/N	No	FLST/RLST trace output

**Table D-12 INC and EXC Statement Diagnostic Parameters**

Parameter	Description
FUNCTION=xxxxxxx	valid MAINVIEW SRM function name (up to eight characters)
JOB=xxxxxxx	job name (up to eight characters)
MODULE=xxxxxxx	valid MAINVIEW SRM module name (up to eight characters)
PGM=xxxxxxx	valid MAINVIEW SRM program name (up to eight characters)
STEP=xxxxxxx	step name (up to eight characters)

# Event Member Parameters

**SMEVNTxx**

SMEVNTxx defines how event notices are to be generated. For more information, see the *MAINVIEW SRM Automation User Guide*.

## Parameter Quick Reference

The following tables provide a brief description of SET statements used in SMEVNTxx, and a brief description of INC and EXC statements used in SMEVNTxx. Detailed descriptions of each parameter are listed in alphabetical order after the tables.

**Table D-13 SET Statement Event Parameters**

Parameter	Required	Description
EVENTID=xxxxx	Yes	Identifies the event
MODE=A/I	Yes	Sets the event to active or inactive
OVERRIDE=Y/N	No	Specifies that default system event parameters are to be replaced
SEV=x	No	Indicates the urgency of the event
TEXT='xxxxx'	Yes	Specifies the text of the event message

**Table D-14 INC and EXC Statement Event Parameters**

Parameter	Required	Description
EVENTID=xxxxx	Yes	Specifies the identifier assigned to the user event in SMEVNTxx.

SMEVNTxx is an optional member.

## Parameter Explanations

### EVENTID=

Purpose:	Specifies an event identifier. The value specified on this parameter is appended to the characters SVW to form an eight-character header for an event message. If a severity indicator is also specified for an event, the indicator will be appended to SVWxxxxx to form a nine-character message header.
Syntax:	<p>EVENTID=xxxxx</p> <p>where xxxxx is a 5 character string. User-defined events cannot begin with the character I, which is reserved for system messages. It is recommended that user-defined events begin with the character U.</p> <p>For user events, this value must match the value assigned on the SET statement EVENTID= parameter in the filter list or rule list member that generates the event.</p> <p>For system events, this value must match the system event ID assigned to the event.</p>
Required:	Yes
Default:	None

### MODE=

Purpose:	Sets an event to active or inactive to turn event generation off or on. If the event mode is inactive, event generation will be bypassed when the function that generates it is processed.
Syntax:	MODE=A/I
Required:	Yes
Default:	System events are defined as inactive. If you want to activate a system event, you must change the value on this parameter to MODE=A.

**Note:** Refreshing an event member reactivates an event.



**VERRIDE=**

**Purpose:** Allows you to replace default values for system events. When **VERRIDE=Y** is specified in an entry, the values you specify on the other parameters in the entry replace the system event default values.

**Syntax:** **VERRIDE=Y/N**

**Required:** No

**Default:** None

**SEV=**

**Purpose:** Indicates the urgency of an event. The severity indicator is appended to the end of **SVWxxxxx** to form a nine-character header for an event message.

**Syntax:** **SEV=x**

where *x* is a single alpha or numeric character. It is recommended that you use the following characters.

I	(informational messages)
W	(warning messages)
E	(error messages)
S	(serious error messages)

**Required:** No

**Default:** None

**TEXT=**

**Purpose:** Specifies the text of the event message.

**Syntax:** **TEXT='xxxxx'**

where the text is enclosed in single quotation marks (') and can contain variables from the function generating the event. The total length of the text can be a maximum of 255 bytes after the variables are expanded. If the text is greater than 255 bytes after variable expansion, it is truncated word by word until it is 255 bytes or less.

Variables used on the TEXT= parameter must be based on INC and EXC statement parameters for functions that generate events. A text variable consists of an ampersand (&) followed by an INC and EXC statement parameter name valid for the function that generates the event. When the event is generated, the value of the parameter is passed to the event and replaces the parameter name in the text.

**Note:** Parameters used as text variables are restricted to INC and EXC statement parameters. You cannot use SET statement-parameter names as variables.

To continue a line of text to the next line, place a non-blank character in column 72 of the line to be continued. The first character in the next line is appended to the last character in the previous line. If you need a blank space to appear in the text following the character in column 72, place a quotation mark (') in the first position of the new line and a space after the quotation mark.

Required: Yes

Default: None

## VTOC Scan Facility Parameters

**SMVSCFxx** SMVSCFxx defines the VTOC Scan Facility filter criteria. For more information, see the *MAINVIEW SRM Reporting User Guide* and the *MAINVIEW SRM Reporting Reference Manual*.

## SVOS VTOC Command

The SVOS VTOC command initiates VTOC scan collection. The output of the collection is written to a sequential data set, where it is available for viewing. The data set name that contains the scan output is indicated in a message appearing in the SVOS job log in response to this command. The output data set is also available in the WBVTOC view.

## Parameter Quick Reference

The following tables provide a brief description of SET statements used in SMVSCFxx and a brief description of INC and EXC statements used in SMVSCFxx. Detailed descriptions of each parameter are listed in alphabetical order after the tables.

SMVSCF<sub>xx</sub> is an optional member.

**Table D-15 SET Statement VTOC Scan Facility Parameters**

Parameter	Required	Description
DSN_MASK=xxxxxxxxxx	N	specifies the data set name or mask
DSN_TYPE=x	N	specifies the data set type
END_UNIT=nnnn	N	specifies the ending unit address range
MASTER=Y/N	N	specifies whether the collected VTOC scan output will go into a completely new collection data set or into the master collection data set
MNT_STATUS=x	N	specifies the volume mount status
MRG_CATINFO=Y/N	N	specifies whether to include catalog information in the collected statistics
MRG_SGCINFO=Y/N	N	specifies whether to include application data in the collected statistics
POOL=xxxxxxxx	N	specifies the user pool mask used to filter volume and data set records processed by VTOC scan collection
RECORD_TYPE=x	N	specifies whether to generate data set or volume records
SGC_APPL=xxxxxxxx	N	specifies the application name mask used to filter volume and data set records processed by VTOC scan collection
SMS_GROUP=xxxxxxxx	N	specifies the SMS group name or mask
SMS_STATE=x	N	specifies the volume's SMS status
START_UNIT=nnnn	N	specifies the starting unit address range
SUBPOOL=xxxxxxxx	N	specifies the subpool mask used to filter volume and data set records processed by VTOC scan collection
VOLUME=xxxxxx	N	specifies the volser or mask

**Table D-16 INC and EXC Statement SMVSCF<sub>xx</sub> Parameter**

Parameter	Required	Description
FORSYSID=	No	Specifies user-defined system IDs that can be included or excluded in a sysplex environment

SMVSCF<sub>xx</sub> is an optional member.

## Parameter Explanations

### **DSN\_MASK=**

Purpose: Specifies the data set name or mask.

Syntax: DSN\_MASK=xxxxxxxxxx

where xxxxxxxx... is the data set name or data set filter. A forward slash specifies all data set names.

Required: No

Default: DSN\_MASK=/'

### **DSN\_TYPE=**

Purpose: Specifies the data set type.

Syntax: DSN\_TYPE=x

where *x* is  
A = All  
V = VSAM  
N = NONVSAM

Required: No

Default: DSN\_TYPE=A

### **END\_UNIT=**

Purpose: Specifies the ending unit address range.

Syntax: END\_UNIT=xxxx

where xxxx is 4 characters

Required: No

Default: END\_UNIT=FFFF

**MASTER=**

**Purpose:** Specifies whether the collected VTOC scan output will go into a completely new collection data set or into the master collection data set.

When MASTER=Yes is specified, the RECORD\_TYPE keyword must be D to collect data set and volume records. Data set filter criteria cannot be specified. This includes the DSN\_MASK, DSN\_TYPE, SGC\_APPL keywords. This is because the data collected must be for all data sets from each qualifying volume. For each volume collected, the old volume data (volume and all its data set records) is replaced with the new data on the master.

When MASTER=No is specified, all filter criteria is available for specification. The collected data is written to a new collection data set.

**Syntax:** MASTER=YES/NO

**Required:** No

**Default:** MASTER=NO

**MNT\_STATUS=**

**Purpose:** Specifies the mount status of the volume.

**Syntax:** MNT\_STATUS=*x*

where *x* is

A = All

P = Public

V = Private

S = Storage

**Required:** No

**Default:** MNT\_STATUS=A

**MRG\_CATINFO=**

**Purpose:** Specifies whether to include catalog information in the collection statistics.

**Syntax:** MRG\_CATINFO=Y/N

**Required:** No

**Default:** MRG\_CATINFO=YES

**MRG\_SGCINFO=**

Purpose: Specifies whether to include application data in the collected statistics.

Syntax: MRG\_SGCINFO=Y/N

Required: No

Default: MRG\_SGCINFO=YES

**POOL=**

Purpose: Specifies the user pool mask used to filter volume and data set records processed by VTOC scan collection.

Syntax: POOL=xxxxxxx

where xxxxxxx is a 1-8 character pool name or mask. A forward slash specifies all pools.

Required: No

Default: POOL=

**RECORD\_TYPE=**

Purpose: Specifies whether to generate the report by data set or volume.

Syntax: RECORD\_TYPE=x

where *x* is

D = data set and volume statistics records  
V = volume statistical records

Required: No

Default: RECORD\_TYPE=V

**SGC\_APPL=**

Purpose: Specifies the application name mask used to filter volume and data set records processed by VTOC scan collection.

Syntax: SGC\_APPL=xxxxxxxxxx

where xxxxxxxxxxx is a 1-50 character application name or mask. A forward slash specifies all applications.

Required: No

Default SGC\_APPL=/  
**SMS\_GROUP=**

Purpose: Specifies the volume's SMS group name or mask.

Syntax: SMS\_GROUP=xxxxxxx

where xxxxxx is an SMS storage group name or filter. A forward slash specifies all SMS storage group names.

Required: No

Default: SMS\_GROUP=/  
**SMS\_STATE=**

Purpose: Specifies the SMS status for the volumes.

Syntax: SMS\_STATE=x

where x is

A= All

I= Initial

M = Managed

U = Unmanaged

Required: No

Default: SMS\_STATE=A  
**START\_UNIT=**

Purpose: Specifies the starting unit address range.

Syntax: START=xxxx

where xxxx is 4 characters

Required: No

Default: START=0000

## **SUBPOOL=**

- Purpose:** Specifies the subpool mask used to filter volume and data set records processed by VTOC scan collection.
- Syntax:** SUBPOOL=xxxxxxx  
 where xxxxxxx is a 1-8 character subpool name or mask. A forward slash specifies all subpools
- Required:** No
- Default:** SUBPOOL=

## **VOLUME=**

- Purpose:** Specifies the volume serial number or volume serial number filter.
- Syntax:** VOLUME=xxxxxx  
 where xxxxxx is 1- to 6-character valid volume serial number defined in your environment. A forward slash specifies all volumes.
- Required:** No
- Default:** VOLUME=



# Appendix E Filter and Rule List Parameters

Table E-1 provides a brief description of INC and EXC and SET statements used in the FLSTxx and RLSTxx members. Following the table is a complete description of each parameter in alphabetical order.

**Note:** <AND> can be used for all INC and EXC parameters that have values of other than Y/N.

**Tip:** Unless otherwise defined, K, M, G, and T (kilobytes, megabytes, gigabytes, and terabytes) can be specified optionally along with a value in numeric parmlib member fields. After the value is converted to bytes, it is checked against system-defined minimum and maximum settings.

For an explanation of how to use filter and rule lists, see the Chapter 5, “Using Filter and Rule List Parameters.”

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 1 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
AA_AMODE=xxxx	X		specifies the application mode: MON, WARN, REJ
AA_APPL=xxxxxxxxxxx	X		specifies the application name (1-50 characters)
AA_ASTAT=xxxx	X		specifies the application status: MDEL, DEL, ACTV
AA_CDATE=xxxxxxxxxxx	X		indicates the date the application was created in the application database in yyyy/mm/dd format
AA_HSMC=nnnnnnnnnn	X		specifies the total amount of space allocated on DASD for HSM data sets assigned to this application (0-9223372036854775807)
AA_HSMH=nnnnnnnnnn	X		indicates the largest amount of space allocated on DASD for HSM data sets assigned to this application (0-9223372036854775807)
AA_KHSM= Y/N	X		indicates if HSM data sets are tracked for this application
AA_KTEMP= Y/N	X		indicates if temporary data sets are tracked for this application
AA_KVSAM= Y/N	X		indicates if VSAM data sets are tracked for this application
AA_LDATE=xxxxxxxxxxx	X		contains the date of the last allocation that resulted in one of the application fields being updated in yyyy/mm/dd format

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 2 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
AA_PERMC=nnnnnnnnnn	X		indicates the total amount of space allocated on DASD for permanent data sets assigned to this application (0-9223372036854775807)
AA_PERMH=nnnnnnnnnn	X		indicates the largest amount of space allocated on DASD for permanent data sets assigned to this application (0-9223372036854775807)
AA_PERMM=nnnnnnnnnn	X		indicates the maximum amount of space allowed for permanent data sets assigned to this application (0-9223372036854775807)
AA_PERMP=nnn	X		percentage of the permanent data set budget currently being used (0-100)
AA_PHSM=Y/N	X		indicates if HSM data set allocations are included as part of the permanent data set allocations
AA_PTEMP=Y/N	X		indicates if temporary data set allocations are included as part of the permanent data set allocations
AA_PVSAM=Y/N	X		indicates if VSAM data set allocations are included as part of the permanent data set allocations
AA_TEMPC=nnnnnnnnnn	X		indicates the total amount of space allocated on DASD for temporary data sets assigned to this application (0-9223372036854775807)
AA_TEMPH=nnnnnnnnnn	X		indicates the largest amount of space allocated on DASD for temporary data sets assigned to this application (0-9223372036854775807)
AA_TEMPM=nnnnnnnnnn	X		specifies the maximum amount of space allowed for temporary data sets assigned to this application (0-9223372036854775807)
AA_TEMPP=nnn			percentage of the temporary data set budget currently being used (0-100)
AA_UFLDn-xxxxxxxxxx	X		1–3 application-defined fields
AA_UNAME=xxxxxxxxxx	X		contains a application-defined name up to 20 characters
AA_VLCNT=nnnnnnnnnn	X		specifies the number of volumes that contain at least one data set included in the application's allocation amounts (0-2147483647)
AA_VSAMC=nnnnnnnnnn	X		indicates the total amount of space allocated on DASD for VSAM data sets assigned to this application (0-9223372036854775807)
AA_VSAMH=nnnnnnnnnn	X		indicates the largest amount of space allocated on DASD for VSAM data sets assigned to this application (0-9223372036854775807)
AA_VSAMM=nnnnnnnnnn	X		indicates the maximum amount of space allowed for VSAM data sets assigned to this application (0-9223372036854775807)
AA_VSAMP=nnn			percentage of the VSAM data set budget currently being used (0-100)
AA_WTHRS=nnn	X		specifies a percentage of the budget that can be used by the application before a warning message is issued (0-100)

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 3 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
AC_CODE=xxxxxxx	X	X	Value in the IBM ACCODE field (8-byte value)
ACF2USER=xxxxxxx	X	X	user name from CA-ACF2 (24-byte value)
ACT_COUNT=nnnnn		X	specifies the maximum to the number of records to which actions can be taken
ACT_EVENTID=xxxxx		X	specifies an event to issue for each record in the SET result group
ACT_JOB=xxxxxxx		X	specifies the name of a member containing skeleton JCL to be submitted using the AutoOPERATOR Skeleton Tailoring facility
ACT_SUM_FLD=fldname		X	causes a running total of the specified field to be maintained for each record against which a specified action is taken
ACT_SUM_LIM= nnnnn		X	limits the number of records to be included in any specified action (0-92233720368)54775807
AD_ALVL1=xxxxxxxxxxxxxxx	X		contains the application name (Application Level 1)
AD_ALVL2=xxxxxxxxxxxxxxx	X		contains the application name (Application Level 2)
AD_ALVL3=xxxxxxxxxxxxxxx	X		contains the application name (Application Level 3)
AD_ALVL4=xxxxxxxxxxxxxxx	X		contains the application name (Application Level 4)
AD_BLKEF=nnn	X		specifies the percentage of blocking efficiency
AD_BLKSZ=nnnnn	X		contains physical block size or VSAM control interval size for the data set
AD_BLKTR=nnnnnnnnnn	X		contains the number of physical blocks that will fit on one track
AD_CASPL=nnnnnnnnnn	X		contains the number of VSAM control area splits performed on the data set
AD_CAT=x	X		contains the catalog status of the data set'
AD_CDATE=xxxxxxxxxx	X		contains the creation date of the data set in yyyy/mm/dd format
AD_CHG=Y/N	X		contains an indicator of whether the data set has been opened for output (changed)
AD_CISPL=nnnnnnnnnn	X		contains the number of VSAM control interval splits performed on the data set
AD_DAYS=nnnnnnnnnn	X		contains the number of days since the data set was opened
AD_DCLAS=xxxxxxx	X		contains the SMS-assigned data class or one of the following values for the data set: NONE, DUPLIC, UNDET
AD_DOCC=nnn	X		contains the occupancy percentage (0-100)
AD_DSN=xxxxxxxxxx	X		specifies the data set name
AD_DSORG=xxxx	X		contains the data set file organization and access method used to manage the data set
AD_EXTS=nnnnnnnnnn	X		contains the number of extents occupied by the data set on the volume
AD_GROUP=xxxxxxxxxx	X		SMS storage group name displayed if the AUTODS function is associated with an AUTOPOOL GROUP= function

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 4 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
AD_LDATE=xxxxxxxxxx	X		contains the last date the data set was opened in yyyy/mm/dd format
AD_LRECL=nnnnn	X		contains the maximum record length for the data set
AD_MCLAS=xxxxxxxx	X		contains the SMS-assigned management class or one of the following values: NONE, DUPLIC, UNDET
AD_POOL=xxxxxxxx	X		pool name displayed if the AUTODS function is associated with an AUTOPool POOL= function
AD_POOLT=S/P	X		contains the pool type, S for subpool or P for user pool
AD_PUSED=nnn	X		contains the percentage of allocation that is used
AD_REBLK=Y/N	X		contains the reblockable indicator, which determines whether the data set can be reblocked by the system when being moved from one device geometry to another
AD_RECFCM=xxxxx	X		contains the data set record format
AD_SCLAS=xxxxxx	X		contains the SMS-assigned storage class or one of the following values: NONE, DUPLIC, UNDET
AD_SIZE=nnnnnnnnnn	X		contains the data set size in kilobytes (one kilobyte equals 1024 bytes) on the volume
AD_SMSI=xx	X		specifies the SMS status of the volume
AD_TRKSA=nnnnnnnnnn	X		contains the number of tracks allocated
AD_TRKSF=nnnnnnnnnn	X		contains the number of tracks unused by the data set on the volume
AD_TRKSU=nnnnnnnnnn	X		contains the number of tracks used by the data set on the volume
AD_VOL=xxxxxx	X		specifies the volume number
AD_VOLSQ=nnnnn	X		specifies the volume sequence number for the data set
AD_XDATE=xxxxxxxxxx	X		specifies the expiration date for the data set in yyyy/mm/dd format
ALCTYPE=xxx	X	X	quantity unit of space allocation (KB, MB, TRK, CYL, KAV, MAV, UAV, BLK) Applies to both primary and secondary space quantities.
ALTPOOL=xxxxxxxx		X	alternate pool for space allocation (1–8 characters)
AP_CTIGC=nnnnnnnnnn	X		specifies the largest contiguous free cylinders (0-2147483647)
AP_CTIGT=nnnnnnnnnn	X		specifies the largest contiguous free tracks (0-2147483647)
AP_FREEEC=nnnnnnnnnn	X		specifies the total count of free cylinders (0-2147483647)
AP_FREED=nnnnnnnnnn	X		specifies the total number of free data set control blocks (DSCB) for all volumes (0-2147483647)
AP_FREET=nnnnnnnnnn	X		specifies the total count of free tracks (0-2147483647)
AP_FREEV=nnnnnnnnnn	X		specifies the total number of free index records in a volume's VTOC index (0-2147483647)
AP_FREEEX=nnnnnnnnnn	X		specifies the total number of free extents (0-2147483647)
AP_FSIZE=nnnnnnnnnn	X		specifies the free size (in MB) of unused space in the pool (0-2147483647)

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 5 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
AP_GROUP=xxxxxxxx	X		specifies the pool, group, or subpool name
AP_HDFUL=nnn	X		specifies the VVDS percentage full high-water mark (1-100)
AP_HFULL=nnn	X		specifies the high-water mark volume percentage full, which is the percentage full of the highest utilized volume in the pool (0-100)
AP_HIFUL=nnn	X		specifies the VTOC percentage full high-water mark (1-100)
AP_HREEC=nnnnnnnnnn	X		specifies the high-water mark count of free cylinders (0-2147483647)
AP_HREED=nnnnnnnnnn	X		specifies the high-water mark count of free DSCBs (0-2147483647)
AP_HREET=nnnnnnnnnn	X		specifies the high-water mark count of free tracks (0-2147483647)
AP_HREEV=nnnnnnnnnn	X		specifies the high-water mark count of free VIR (0-2147483647)
AP_HREEX=nnnnnnnnnn	X		specifies the high-water mark count of free extent (0-2147483647)
AP_HVFRG=nnn	X		specifies the high-water mark fragmentation index (0-100)
AP_HVFUL=nnn	X		specifies the high-water mark VTOC percentage full (0-100)
AP_LDFUL=nnn	X		specifies the VVDS percentage full low-water mark (1-100)
AP_LFULL=nnn	X		specifies the low-water mark volume percentage full (0-100)
AP_LIFUL=nnn	X		specifies the VTOC percentage full low-water mark (1-100)
AP_LPRIC=nnnnnnnnnn	X		specifies the largest primary allocation (cylinders) (0-2147483647)
AP_LPRIT=nnnnnnnnnn	X		specifies the largest primary allocation (tracks) (0-2147483647)
AP_LREEC=nnnnnnnnnn	X		specifies the low-water mark free count of cylinders (0-2147483647)
AP_LREED=nnnnnnnnnn	X		specifies the low-water mark free count of DSCBs (0-2147483647)
AP_LREET=nnnnnnnnnn	X		specifies the low-water mark free count of tracks (0-2147483647)
AP_LREEV=nnnnnnnnnn	X		specifies the low-water mark free count of VIR (0-2147483647)
AP_LREEX=nnnnnnnnnn	X		specifies the low-water mark free count of extent (0-2147483647)
AP_LVFRG=nnn	X		specifies the low-water mark fragmentation index (0-100)
AP_LVFUL=nnn	X		specifies the low-water mark of VTOC percentage full
AP_PERFL=nnn	X		specifies the pool percentage full based on used space as it relates to total space (0-100)
AP_POOL=xxxxxxxx	X		specifies the pool, group, or subpool name (1-30)

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 6 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
AP_TSIZE=nnnnnnnnnn	X		specifies the total size (in MB) of space in the pool (0-2147483647)
AP_TYPE=xxxxxxx	X		specifies the pool type: POOL, SMSPOOL, or GROUP
AP_USIZE=nnnnnnnnnn	X		specifies the amount of allocated space in the pool (used size) (0-2147483647)
AP_VOLC=nnnnnnnnnn	X		specifies the number of online volumes in this pool on the collecting OS/390 system (0-2147483647)
AP_VOLD=nnnnnnnnnn	X		specifies the volume drop count (due to errors) (0-2147483647)
AUTOLEV=xxxxxxx	X		contains an 8-character literal AUTOLEVx, where x is a number indicating the current automation level for the resource being automated
AV_CTIGC=nnnnnnnnnn	X		contains the largest single extent in full cylinders available for allocation (0-2147483647)
AV_CTIGT=nnnnnnnnnn	X		contains the largest single extent in tracks available for allocation (0-2147483647)
AV_DEV=xxxxxxx	X		contains the generic unit name for the volume (1-8 characters)
AV_FRAGI=nnnnnnnnnn	X		contains the fragmentation index value of the volume (0-2147483647)
AV_FREEC=nnnnnnnnnn	X		contains the number of free cylinders on the volume (0-2147483647)
AV_FREED=nnnnnnnnnn	X		contains the number of free (Format 0) DSCBs on the volume (0-2147483647)
AV_FREET=nnnnnnnnnn	X		contains the number of free tracks on the volume (0-2147483647)
AV_FREEV=nnnnnnnnnn	X		contains the number of free VIRs (VTOC index records) on the volume (0-2147483647)
AV_FREEX=nnnnnnnnnn	X		total amount of free extents on the volume (0-2147483647)
AV_FSIZE=nnnnnnnnnn	X		number of tracks not used on the volume (0-2147483647)
AV_FULL=nnn	X		contains the percentage of used space to total space for the volume (0-100)
AV_IPCTF=nnn	X		specifies the VTOC index percentage full between 0 and 100
AV_ISIZE=nnnnnnnnnn	X		specifies the total size of VVDS (0-2147483647)
AV_ISIZE=nnnnnnnnnn	X		specifies the total size of the VTOC index (0-2147483647)
AV_LPRIC=nnnnnnnnnn	X		contains the largest possible primary extent in cylinders; the sum of the 5 largest extents on the volume (0-2147483647)
AV_LPRIT=nnnnnnnnnn	X		contains the largest possible primary extent in tracks; the sum of the 5 largest extents on the volume (0-2147483647)
AV_MNT=xxx	X		contains how the volume is mounted: PUB Public, PVT Private, STG Storage, SYS System
AV_PCNT=n	X		specifies the pool count (1-8)

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 7 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
AV_PID=xxxxxxxxxx	X		specifies the physical disk ID (1-11 characters)
AV_POOLn=xxxxxxxx	X		specifies the pool name (1-8 characters) for pool number 1-8
AV_PTYN=S/P	X		specifies S for subpool or U for user pool for pool type 1-8
AV_SMSGP=xxxxxxxx	X		contains the SMS -assigned storage group name
AV_SMSI=xx	X		contains the SMS status of the volume: M = SMS managed, QA = SMS quiesced all, QN = SMS quiesced new, DA = SMS disabled all, DN = SMS disabled new, UN = Not SMS managed, NA = Unknown
AV_TSIZE=nnnnnnnnnn	X		total volume size in tracks (1-8 characters)
AV_UCB=hhhh	X		contains the device number for the volume (4 hexadecimal digits)
AV_USIZE=nnnnnnnnnn	X		number of tracks used on the volume (0-2147483647)
AV_VOL=xxxxxx	X		contains the volume serial number (1-6 characters)
AV_VPCTF=nnn	X		specifies the VVDSF percentage full between 0 and 100
AV_VTOCF=nnn	X		contains the percentage of used VTOC space (0-100_
AV_VTOCI=xxx	X		contains the VTOC index status: ACT, INA, UND, DIS
AV_VTOCZ=nnnnnnnnnn	X		contains the volumes VTOC size in tracks (0-2147483647)
AVL=nnnnn		X	average block or record length (1–32,767)
BACKCMD= Y/N		X	DFHSM backup on command
BACKUP= Y/N		X	DFHSM ML0-ML2 migration without backup
BLKSIZE=<>nnnnn	X	X	block size of data set (0–32,760)
BUFSP=nnnnnn	X	X	buffer space for VSAM data sets (0–16776704)
CAL= Y/N		X	adjust date by non-working day calendar
CALAGE=nnnn	X		calendar-adjusted unreferenced day count (0–9999)
CANDIDATE= Y/N		X	candidate volume accepted
CAT=xxxxxxxxxxxxxx	X		catalog name (1–44 characters)
CATALOG= Y/N	X	X	allows or removes IDCAMS CATALOG parameter
CISIZE=(nnnnnn,nnnnnn)	X	X	VSAM data and/or index control interval size (0–999999)
COMP= Y/N		X	cartridge tape data set compression
CONTIG= Y/N	X	X	specifies whether a data set is allocated with contiguous space required
CRITBIAS=nnn		X	defines the number of data sets that can reside on the volume before the current allocation
CRITEMC= Y/N		X	specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes
CRITFAIL= Y/N		X	defines the allocation if a volume meeting the criteria cannot be found
CRITLIST=xxxxxxxx		X	name of the table containing the allocation volumes
CURDAY=xxxxxxxxxx	X		current day of the week (1–10 characters)
CURSPACE=nnnnnnK,M,G,T	X		current size of data set (0–999999K,M,G,T)
CURTIME=nn:nn:nn	X		current time expressed as HH:MM:SS

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 8 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
DADSM_FUNC=xxxxxxxx,xxxxxxxx, ...	X		current location in the allocation process (JCL, ALLOCATE, EXTENDNV, RENAME, VOLSEL)
DATACLAS=xxxxxxxx	X	X	DFSMS Data Class name (1–8 characters)
DD=xxxxxxxx	X		data definition statement name (1–8 characters)
DEFUNIT=xxxxxxxx		X	generic unit name for volumes outside the silos
DEVTYPE=xxxx	X		device type (DASD, 3380, 3390, TAPE, UNKN)
DIR=nnnn	X	X	sets number of directory blocks for partitioned data sets (1–9999)
DISPn=xxxxxx	X		data set disposition, n=1-3 (NEW, KEEP, and so on)
DPORDEF=nnn		X	default response time for device selection (0-999)
DPORMAX=nnn		X	maximum response time target for device selection (0-999)
DPORMIN=nnn		X	minimum response time target for device selections (0-999)
DPORSEP=nnn		X	separation factor for device selection (0-999)
DPOWIND=nnnn		X	interval for device selection based on performance (0-999)
DSN=xxxxxxxxxxxxxxxxxx	X		data set name (1–44 characters)
DSNAME=xxxxxxxxxxxxxxxxxx	X		synonym (see DSN)
DSNn=xxxxxxxx	X	X	data set name qualifier, n=1–8 (1–8 characters)
DSNTYPE=xxx	X		data set name type (PDS, LIB, HFS, PIP, DB2®, IAM, STR) Will not be set to IAM during NOCATLG2, SPACSWIR, or SPACPRIM processing.
DSORG=xx	X		data set organization (PS, PO, IS, VS, DA, PDSE, --)
DSTYPE=xxxx	X		data set type (PERM, TEMP, GDG)
DYNALLOC= Y/N	X		allocation is dynamic
ENVIR=xxxxxxxx	X		DFSMS allocation environment
ERASE= Y/N	X	X	allows or removes IDCAMS ERASE parameter
EVENTID=xxxxx	X	X	specifies an event identifier
EXPDT=nnnnnn	X	X	expiration date for a data set (90001–99365)
EXPDT=nnnnnnnn	X	X	expiration date for a data set (yyyyddd)
EXTENT=<>nnn	X		number of extents (1–123)
FILESEQ=nnnnnn	X		file sequence number (0–999999)
FORCE= Y/N		X	overrides program specified blocksize
FUNCTION=xxxxxxxx	X		specifies the name of the current function
GDGVER=<>nnn	X		contains the relative version number (0–255), relative to the current generation of a GDG data set
HDPODAYS=(MO,TU,WE,TH FR,SA,SU)		X	days of the week used in determining performance statistics for pooling
HDPOETIM=nnnn		X	ending time each day for a range of snapshots to be analyzed (0-2359)
HDPORDEF=nn		X	default response time for device selection (0-999)
HDPORMAX=nn		X	maximum response time target for device selection (0-999)



**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 9 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
HDPORMIN= <i>nn</i>		X	minimum response time target for device selection (0-999)
HDPORSEP= <i>nn</i>		X	separation factor for device selection (0-999)
HDPOSTIM= <i>nnnn</i>		X	starting time each day for a range of snapshots to be analyzed (0-2359)
HLQ= <i>xxxxxxxx</i>	X		HLQ of data set name (1–8 characters)
HSM= Y/N	X		flags a DFHSM migration or backup data set
HSMSDN= <i>xxxxxxxx</i>	X		specifies the DFHSM migration or backup data set name (1–44 characters)
IMBED= Y/N	X	X	allows or removes IDCAMS IMBED parameter
JOB= <i>xxxxxxxx</i>	X		job, TSO, or STC name (1–8 characters)
JOBACCTn= <i>xxxxxxxxxxx</i>	X		job account field, n=1–3 (1–20 characters)
JOBCLASS= <i>x</i>	X		job class (1 character)
JOBSDAY= <i>xxxxxxxx</i>	X		job start day (1–8 characters)
JOBSTIME= <i>nn:nn:nn</i>	X		job start time expressed as HH:MM:SS
JOBTYPE= <i>STC/TSO/JOB</i>	X		specifies the type of job that issued a request (STARTED TASK/TIME SHARING USER/BATCH JOB)
LABELTYP= <i>xxx</i>	X		determines tape label characteristics
LEVEL= <i>n</i>	X		specifies the account level being processed by the application collector (a value from 1–4)
LIMIT= <i>nnnnnnnnK,M,G,T</i>		X	data set size limit in K,M,G,T (1–99,999,999)
LLQ= <i>xxxxxxxx</i>	X	X	LLQ of data set name (1–8 characters)
LRECL= <i>&lt;&gt;nnnnnn</i>	X		logical record length (1–32,760)
MAXQLF= <i>nn</i>		X	maximum number qualifiers in data set name (1–99)
MAXSIZE= <i>&lt;&gt;nnnnnnnnnnK,M,G,T</i>	X		maximum data set size 1–2147483647K
MGMTCLAS= <i>xxxxxxxx</i>	X	X	DFSMS Management Class (1–8 characters)
MIGCMD= Y/N		X	DFHSM migration on command
MIGDAYS= <i>nnnn</i>		X	elapsed days until migration (1–9999) (No longer valid for HSM MIGRT)
MIGRATE= Y/N		X	migration permission to DFHSM
MINQLF= <i>nn</i>		X	minimum number qualifiers in data set name (1–99)
ML2= Y/N		X	allow direct DFHSM migrate from ML0 to ML2
MNTYPE= <i>((mmmmmm,n,op),.)</i>	X	X	mount type triplets used for compatibility with STOP-X37
MNTYPE= <i>xxxxxxxx</i>	X	X	mount type (PUBLIC, STORAGE, PRIVATE, CURRENT, ALL)
MODE= <i>ACT/INACT/SIM</i>	X		sets the status of the function
MSG= <i>I/W/E/S/N</i>	X		level of messages to be generated
NEWAPPL= <i>xxxxxxxxxxxxxxxx</i>		X	specifies the value of an account code that is used to override the default account code
NOCATLG2= <i>xxxxxxx</i>		X	action on a not cataloged 2 condition
NOCATWHEN= <i>TERM/ALLOC</i>		X	specifies when NOCATLG2 processing will occur for a non-SMS data set
NOCHECK= <i>xxxxxxx</i>		X	specifies checks bypassed in volume switch validation

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 10 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
NQUAL=<>nn	X		number of qualifiers in data set name (1–20)
NUNIT=nn	X		number of units requested (1–59)
NVOL=nn	X	X	number of volumes that can be allocated to or requested for a non-VSAM data set (1–59) or VSAM data component (1–20)
NVOLINDX=nn		X	number of volumes that can be allocated to a VSAM index component (1–20)
NVOLMAX= Y/N		X	maximum number of volumes allocated to a data set
OLDACCT=xxxxxxxxxxxxx	X		specifies the value of the default account code (1–50 characters)
OLDDSN=xxxxxxxx	X		old data set name (1–44)
OLDHLQ=xxxxxxx	X		old data set HLQ (1–8 characters)
OPER= Y/N		X	allows operator to provide volume when system cannot find space for a volume switch
ORIGUNIT=xxxxxxx	X		unit specified in the JCL 1–8 characters long; cannot be set for SPACPRIM, SPACSECA, SPACSECB, SPACSECR, SPACSWIR, or SPACVOLA
ORIGVOL=xxxxxx	X		volume specified in the JCL. 1–6 characters long
OWNER=xxxxxxxxxxxxx	X	X	assigns an owner ID to a VSAM cluster (1–40 characters); cannot be set for NOCATLG2, SPACSWIR, or SPACPRIM functions
PCTI=nnnnn		X	specifies the percentage value by which a secondary allocation is increased (1-10000)
PGM=xxxxxxx	X		program name (1–8 characters)
PGMRNAME=xxxxxxxxx	X		programmer name job card field (1–20 characters)
POOL=xxxxxxx	X	X	1–15 pool names (1–8 characters)
PQTY=nnnnnK,M,G,T		X	primary quantity space allocation (1–99,999 K,M,G,T)
PRISPACE=nnnnnn	X		primary space requested (0–999999)
PROCSTEP=xxxxxxx	X		procedure step name (1–8 characters)
PURGE= Y/N		X	purge a deleted data set with an expiration date
PWDEL= Y/N		X	delete passwords specified in control statements
QUALL=xxxxxxx	X		synonym (see LLQ)
QUALn=xxxxxxx	X		synonym (see DSNn)
RACF=xxxxxxx	X		RACF group name (1–8 characters)
RACFGRP=xxxxxxx	X		RACF group name (1–8 characters)
RACFUID=xxxxxxx	X		specifies the value of the RACF user ID on a JOB card (1–8 characters)
RAIDDEVTYPE=xxxxxxx	X		allows the user to require a specific RAID device type for an allocation (EMC/RDFEMC/MIRROREMC/PARITYEMC)
RECFM=xxx	X		record format
REORG=xx	X		VSAM record organization (RR, ES, KS, LS)
REFAGE=nnnn	X		unadjusted unreferenced day count (0–9999)
REFVOL=xxxxx	X		DASD volser from VOL=REF=

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 11 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
REJECT= Y/N		X	rejects an OS/390 request for a specific service
RELEASE= Y/N	X		release flag
REORG_NSMS=(xx,pool)		X	SMRORGxx member name suffix in parmlib for DFDSS reorg job control cards; name of pool to which MAINVIEW SRM is to reorganize
REORG_PROC=xxxxxxx		X	reorganize proc name for SPACVOLA to start
REORG_SMS=(xx,storclas)		X	SMRORGxx member name suffix in parmlib for DFDSS reorg job control cards; DFSMS Storage Class to which MAINVIEW SRM is to reorganize
REORG= Y/N		X	indicates whether SPACVOLA processing should automatically start a started task to reorganize the file that was just made multivolume by SPACVOLA
REPL= Y/N	X	X	allows or removes IDCAMS REPLICATE parameter
REPLACE= Y/N		X	allows replacement of system value
RETPD=nnnn	X	X	retention period for data set (0–9999)
REUSE= Y/N	X	X	allows or removes IDCAMS REUSE parameter
RLSE=ALL/SEC/NO		X	data sets eligible for space release
ROUND= Y/N	X	X	round space to cylinders
SCAN=EXIT		X	specifies not to budget space for any level associated with a data set; this parameter is unique to the application collector
SECSpace=nnnnnnK,M,G,T	X		secondary space requested (0–999999K,M,G,T)
SEP= Y/N/ASIS		X	specifies whether the data and index components of a VSAM key-sequenced data set are allocated to separate volumes in a pool
SGC_FUNC=xxxxxxx	X		specifies the value of the application function currently being processed (ALLOCATE/EXTENDCV/EXTENDNV/EXTENDVS/RELEASE/RENAME/SCRATCH/BUDGET/BUDDSN/SGCMAINT/SGCRSYNC/SGCHSMR/SVOSISPF)
SGDA_ALNV=nnnnnnnnnn	X		specifies the total space allocated to non-VSAM data sets in the account
SGDA_ALV=nnnnnnnnnn	X		specifies the total space allocated to VSAM data sets in the account
SGDA_AVAIL=nnnnnnnnnn	X		specifies the total space available in the account
SGDA_GRP=xxxxxxx	X		specifies the application group name; also known as account name
SGDA_IDLE=nnnnnnnnnn	X		specifies the total allocated space that is unused in the account
SGDA_NVDS=nnnnn	X		specifies the number non-VSAM data sets in the account
SGDA_VSD=nnnnn	X		specifies total number of VSAM data sets in the account
SGDP_ALNV=nnnnnnnnnn	X		specifies the space allocated for non-VSAM data sets in the pool
SGDP_ALV=nnnnnnnnnn	X		specifies the total space allocated to VSAM data sets in the pool
SGDP_AVAIL=nnnnnnnnnn	X		specifies the total space available in the pool

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 12 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
SGDP_DS#DN=nnnnnnnn	X		specifies the total number of data sets on volumes with a SMS status of DISABLED/NEW
SGDP_DS#EN=nnnnnnnn	X		specifies the total number of data sets on volumes with an SMS status of ENABLED
SGDP_DS#NM=nnnnnnnn	X		specifies the total number of data sets on volumes with an SMS status of NOT SMS MANAGED
SGDP_DS#NS=nnnnnnnn	X		specifies the total number of data sets on volumes with an SMS status of NO STATUS GIVEN
SGDP_DS#QA=nnnnnnnn	X		specifies the total number of data sets on volumes with an SMS status of QUIESCED/ALL
SGDP_DS#QN=nnnnnnnn	X		specifies the total number of data sets on volumes with an SMS status of QUIESCED/NEW
SGDP_FRAGI=nnnn	X		specifies the fragmentation index.
SGDP_IDLE=nnnnnnnnnn	X		specifies the space allocated and unused in the pool
SGDP_NCLPER=nnnn	X		specifies the net capacity load percentage in tenths of a percent (0-1000)
SGDP_NNV=nnnnn	X		specifies the number of non-VSAM data sets in the pool
SGDP_NV=nnnnn	X		specifies the number of VSAM data sets in the pool
SGDP_NVOL=nnnnn	X		specifies the number of volumes in the pool
SGDP_OFFL=nnnnn	X		specifies the total offline volumes offline
SGDP_PERFUL	X		specifies the Percentage Full or Percentage Allocated for all volumes in the pool
SGDP_POOL=xxxxxxx	X		specifies the pool name for reporting
SGDP_RSVD=nnnnnnnnnn	X		specifies the total reserved space in the pool
SGDP_RVAARC=nnnnnnnnnn	X		specifies the array capacity of the device for RVA pools
SGDP_RVAFNC=nnnnnnnnnn	X		specifies the amount of space not collected by free space collection activity during the interval for RVA pools
SGDP_RVAFSC=nnnnnnnnnn	X		specifies the amount of space collected by free space collection activity during the interval for RVA pools
SGDP_RVAIND=Y/N	X		specifies whether the pool is for an RVA device
SGDP_RVANCL=nnnnnnnnnn	X		specifies the net capacity load of the RVA device
SGDP_SPACDA=nnnnnnnnnn	X		specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of DISABLED/ALL
SGDP_SPACDN=nnnnnnnnnn	X		specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of DISABLED/NEW
SGDP_SPACEN=nnnnnnnnnn	X		specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of ENABLED
SGDP_SPACNM=nnnnnnnnnn	X		specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of NOT SMS MANAGED
SGDP_SPACNS=nnnnnnnnnn	X		specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of NO STATUS GIVEN
SGDP_SPACQA=nnnnnnnnnn	X		specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of QUIESCED/ALL

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 13 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
SGDP_SPACQN=nnnnnnnnnn	X		specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of QUIESCED/NEW
SGDP_TYPE=x	X		specifies the type of pool
SGDP_VOL#DA=nnnnn	X		specifies the total number of volumes with a SMS status of DISABLED/ALL
SGDP_VOL#DN=nnnnn	X		specifies the total number of volumes with an SMS status of DISABLED/NEW
SGDP_VOL#EN=nnnnn	X		specifies the total number of volumes with an SMS status of ENABLED
SGDP_VOL#NM=nnnnn	X		specifies the total number of volumes with an SMS status of NOT SMS MANAGED
SGDP_VOL#NS=nnnnn	X		specifies the total number of volumes with an SMS status of NO STATUS GIVEN
SGDP_VOL#QA=nnnnn	X		specifies the total number of volumes with an SMS status of QUIESCED/ALL
SGDP_VOL#QN=nnnnn	X		specifies the total number of volumes with an SMS status of QUIESCED/NEW
SGDV_ALREXT=nnnnn	X		specifies the number of additional tracks in largest free extent on the volume
SGDV_FRAGI=nnnnn	X		specifies the fragmentation index on the volume
SGDV_FRCYL=nnnnn	X		specifies the number of free cylinders on the volume
SGDV_FREXT=nnnnn	X		specifies the number of free extents on the volume
SGDV_FRVIR=nnnnn	X		specifies the free VIR count on the volume
SGDV_IDTR=nnnnn	X		specifies the total number of idle tracks on the volume
SGDV_LREXT=nnnnn	X		specifies the number of cylinders in largest free extent on the volume
SGDV_LREXTT=nnnnn	X		specifies the size of largest extent in tracks on the volume
SGDV_NDS=nnnnn	X		specifies the total number of data sets on the volume
SGDV_NF0DSC=nnnnn	X		specifies the format 0 (free) DSCB count on the volume
SGDV_PERFUL	X		specifies the Percentage Full or Percentage Allocated for the volume
SGDV_PHYID=xxxxxxxxxx	X		specifies the physical disk ID
SGDV_PHYIDT=x	X		specifies the physical disk ID type
SGDV_POOL=xxxxxxx	X		specifies the first pool name in which the volume is defined
SGDV_POOL1=xxxxxxx	X		specifies pool name in which the volume is defined
SGDV_PTYP=x	X		specifies the pool type
SGDV_RSRVDT=nnnnn	X		specifies the number of reserved tracks (not included in free space) on the volume
SGDV_RVAFDV=xx	X		specifies the functional device ID for a volume existing on a RVA frame
SGDV_RVAIND= Y/N	X		indicates whether the volume exists on a RVA frame
SGDV_RVAPCS=nnnnn	X		specifies the physical capacity shared for a volume existing on a RVA device

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 14 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
SGDV_RVAPCU=nnnnn	X		specifies the physical capacity used for a volume existing on a RVA device
SGDV_RVASSF=xxxxxxxx	X		specifies the RVA subsystem frame name for the RVA frame the volume exists on
SGDV_RVAVOL=xxxxxxxx	X		specifies the descriptive volume name of a volume existing on a RVA frame
SGDV_SMALLD=nnnnnnnn	X		specifies the size of the smallest data set on the volume (in kilobytes)
SGDV_SSID=xxxx	X		specifies the subsystem ID
SGDV_USEXT=nnnnn	X		specifies the number of used extents on the volume
SGDV_VIRPU=nnn	X		specifies the VTOC index percentage used
SGDV_VIRSZ=nnnnnnnnnn	X		specifies the VTOC index size in tracks
SGDV_VOL=xxxxxxxx	X		specifies the volume serial number of the volume
SGDV_VVDSPU=nnn	X		specifies the VVDS percentage used
SGDV_VVDSSZ=nnnnnnnnnn	X		specifies the VVDS size in tracks
SGP_@BUSY=><nnn	X		specifies channel path busy threshold for inclusion or exclusion
SGP_BESCOLT=nnnnnnnn	X		specifies the collected back-end space in tenths of a MB
SGP_BESFREE=nnnnnnnn	X		specifies the free back-end space in tenths of a MB
SGP_BESTOTL=nnnnnnnn	X		specifies the total back-end space in tenths of a MB
SGP_BESUNCL=nnnnnnnn	X		specifies the uncollected back-end space in tenths of a MB
SGP_CFWHIT@=><nnn	X		specifies percentage of DFAST reads satisfied by cache threshold
SGP_CFWPRSC=><nnn	X		specifies number of CFAST writes reads per-second threshold
SGP_CHPID=><xx	X		specifies channel paths to be included or excluded
SGP_CNTLUID=><xx	X		specifies subsystem IDs of cache controllers to be included or excluded
SGP_CONNTIM=><nnnnn	X		specifies the data set connect time threshold in .1 millisecond increments
SGP_CUBSYDL=><nnnnn	X		specifies the control unit busy delay threshold in .1 millisecond increment
SGP_DFWHIT@=><nn	X		specifies percentage of DFAST writes satisfied by cache threshold
SGP_DFWPRSC=><nnn	X		specifies number of DFAST writes per-second threshold
SGP_DISCTIM=><nnnnn	X		specifies the data set disconnect time threshold in .1 millisecond increments
SGP_DP@BUSY=><nn	X		specifies director port busy percentage to be included or excluded
SGP_DPBSYDL=><nnnnn	X		specifies the director port busy delay time threshold in .1 millisecond increments
SGP_DVBSYDL=><nnnnn	X		specifies the device busy delay time threshold in .1 millisecond increments

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 15 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
SGP_ECMCFBS=nnnnnnnn	X		specifies the ECAM channel programs bypassed due to busy configuration in tenths of a MB
SGP_ECMMSGs=nnnnnnnn	X		specifies ECAM messages processed in tenths of a MB
SGP_ECMNSPC=nnnnnnnn			specifies the ECAM channels programs bypassed due to no buffer space in tenths of a MB
SGP_ECMPGMS=nnnnnnnn	X		specifies the ECAM channel programs in tenths of a MB
SGP_FSCBYRD=nnnnnnnn	X		specifies the collected free space bytes read in tenths of a MB
SGP_FSCPERC=nnnn	X		specifies the percentage of collected free space in tenths of a percent
SGP_FSUPERC=nnnn	X		specifies the net capacity load percentage in tenths of a percent
SGP_IOPRSEC=><nnn	X		specifies number of IOs per-second threshold
SGP_IOSQTIM=><nnnnn	X		specifies the data set IOSQ time threshold in .1 millisecond increments
SGP_LCU@BUSY=><nn	X		specifies LCU busy percentage to be included or excluded
SGP_LCUID=><xx	X		specifies the logical control unit id of those controllers to be included or excluded
SGP_NCLPERC=nnnn	X		specifies the percentage of uncollected free space in tenths of a percent
SGP_NRDHIT@=><nnn	X		specifies percentage of normal reads satisfied by cache threshold
SGP_NRDPRSC=><nnn	X		specifies number of normal reads per-second threshold
SGP_NWRHIT@=><nnn	X		specifies percentage of normal writes satisfied by cache threshold
SGP_NWRTPSC=><nnn	X		specifies number of normal writes per-second threshold
SGP_PENDTIM=><nnnnn	X		specifies the data set pending time threshold in .1 millisecond increments
SGP_RDHIT@=><nnn	X		specifies percentage of reads satisfied by cache threshold
SGP_RDSPRSC=><nnn	X		specifies number of reads per-second threshold
SGP_READ@=><nnn	X		specifies the percentage of IOs that are reads threshold
SGP_RESERV@=><nn	X		specifies percentage volume is reserved for inclusion or exclusion
SGP_RESPTIM=><nnnnn	X		specifies the data set response time threshold in .1 millisecond increments
SGP_RSFNAME=xxxxxxxx	X		specifies the IXPf subsystem frame name
SGP_SRDHIT@=><nnn	X		specifies percentage of sequential reads satisfied by cache threshold
SGP_SRDPRSC=><nnn	X		specifies number of sequential reads per-second threshold
SGP_SWRHIT@=><nnn	X		specifies percentage of sequential writes satisfied by cache threshold
SGP_SWRPRSC=><nnn	X		specifies number of sequential writes reads per-second threshold

**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 16 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
SGP_WRHIT@=><nnn	X		specifies percentage of writes satisfied by cache threshold
SGP_WRITE@=><nnn	X		specifies percentage of IOs that are writes threshold
SGP_WRPSEC=><nnn	X		specifies number of writes per-second threshold
SIZE=<>nnnnnnnnnnK,M,G,T	X		size of either primary extent or of primary + 2 secondary extent, 1–2147483647K
SMF=I/W/E/S/N	X		level of messages to be generated to SMF
SMS=Y/N		X	synonym (see SMSMANAGED)
SMSMANAGED=Y/N	X	X	specifies whether a resource is managed by DFSMS
SMSPOOL_EXT=(xxxxxxxx,xxxxxx xx,...)		X	defines an SMSPOOLS to be used during DADSM EXTEND for SMS-managed data sets
SMSPOOL=(xxxxxxxx,xxxxxxx,...)		X	defines an SMSPOOLS to be used during DADSM ALLOCATE for SMS-managed data sets
SOLUTION=xxxxxxxx	X		contains the solution value from the originating AUTO function command (1-8 characters)
SORT=(fldname,x,fldname,x...)		X	specifies the fields to sort prior to taking any actions on the group
SPACPRIM=(nn,nn)		X	lower limit and decrement percentage for space reduction (0–90)
SPACSECA=nnnn		X	secondary space as a percentage of primary (1–9999)
SPACSECB=nn		X	lower limit for space reduction (0–100)
SPACSECI=nn		X	extent limit, secondary space enlargement (1–15)
SPACSECR=nnn		X	specifies a percentage and floor limit for space reduction
SPACSWIR=(nnn,nnn)		X	specifies the lower limit and decrement of space reduction (0–100)
SPACVOLA=nn		X	maximum volumes to extend a data set (1–59) (does not support SAS data libraries)
SPECIFIC=Y/N	X		specific or non-specific volume specification
SPLIT=Y/N		X	specifies whether to split unit affinities with STK silos
SQTY=nnnnnnK,M,G,T		X	secondary quantity space allocation (1–99999K,M,G,T)
STEP=xxxxxxxx	X		jobstep name (1–8 characters)
STEPACCTn=xxxxxxxxxxx	X		step account field, n=1–3 (1–20 characters)
STOGROUP=xxxxxxxx	X	X	DFSMS Storage Group of data set (1–8 characters)
STORCLAS=xxxxxxxx	X	X	DFSMS Storage Class of data set (1–8 characters)
STORGRP		X	synonym (see STOGROUP)
STRIPCNT=nnnnnnnn	X		determines the number of stripes the data set has (1–99999999)
STRIPTY=SS/SM/VS	X		type of extended format data set
SUPVOL=Y/N		X	suppresses requests for specific volumes
SYSID=xxxx	X	X	OS/390 system ID (1–4 characters)
TEMPDSN=Y/N	X		flags temporary data sets
TRKCYL=nn		X	tracks per cylinder source for SPACCONV
TRKLEN=nnnnn		X	bytes per track of source for SPACCONV



**Table E-1 Filter and Rule List Parameter Quick-Reference (Part 17 of 17)**

Parameter	INC and EXC FLST/RLST	SET RLST	Description
UNIT=xxxxxxxx	X	X	unit name (generic or esoteric) 1–8 characters long
USECPOOL= Y/N		X	search current pool first for an additional volume
USER=xxxxxxxx	X		user name (1–8 characters)
USEVOL=xxxx		X	directs volume allocation to STOR, PRIV, ALL
USRCn=xxxxxxxx	X	X	character field for a user-specified variable 1–8 characters long; the value of <i>n</i> can be 1–10 (for example USRC1, USRC2, and so forth)
USRNy=nnn	X	X	numeric field for a user-specified variable not to exceed 214783647; the value of <i>y</i> can be 1–10 (for example USRN1, USRN2, and so forth)
VCOMPLLQ=xxxxxxxx	X	X	LLQ of VSAM component (1–8 characters)
VFORCE= Y/N		X	adds standard component suffixes (DATA, INDEX) to VSAM file names
VIO= Y/N		X	directs data sets to VIO
VOL=xxxxxx	X	X	volume name (1–6 characters)
VOLSEL=xxxxxxxx		X	volume selection criteria
VOLSER=((xxxxxx,n,op),.)		X	volume serial ID triplets used for compatibility with STOP-X37
VOLSER=xxxxxx		X	volume serial ID (1–6 characters)
VSAMCOMP=xxxxx	X		VSAM data set comp type (DATA, INDEX)
VSAMDEF=xxxxxxx	X		VSAM data set cluster definition
VSAMSEP= Y/N	X		indicates data and index components are on separate volumes
XMODE=STC/TSO/JOB	X		job execution mode

## Parameter Explanations

### **AA\_AMODE=**

Purpose: Specifies the application mode.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_AMODE=xxxx

Valid values are:

MON Tracks space allocations and deallocations as they occur allowing up-to-the-minute analysis of DASD space usage.

WARN A message is generated if the current allocation exceeds the budget amount.

REJ If the current allocation exceeds the budget amount, the allocation will be rejected.

### **AA\_APPL=**

Purpose: Specifies the application name.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_APPL=xxxxxxxxxx

where xxxxxxxxxxxx is the application name up to 50 characters

If there is a blank in the application name, enclose the name in quotes.

**AA\_ASTAT=**

Purpose: Indicates the application status.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_ASTAT=xxxx

Valid values are:

**MDEL** The application has been manually flagged for deletion. The next time that the database is copied, this application will be deleted.

**DEL** This application has been automatically flagged for deletion. This application was created, but never updated. Because no activity has taken place in the application, it will be deleted the next time the database is copied.

**ACTV** This is a currently active application.

**AA\_CDATE=**

Purpose: Indicates the date the application was created in the application database in yyyy/mm/dd format.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_CDATE=xxxxxxxxxx

**AA\_HSMC=**

Purpose: Indicates the total amount of space allocated on DASD for HSM data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_HSMC=nnnnnnnnnn

where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_HSMH=**

Purpose: Indicates the largest amount of space allocated on DASD for HSM data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_HSMH=nnnnnnnnnn

where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_KHSM=**

Purpose: Indicates if HSM data sets are tracked for this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_KHSM=Y/N

**AA\_KTEMP=**

Purpose: Indicates if temporary data sets are tracked for this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_KTEMP=Y/N

**AA\_KVSAM=**

Purpose: Indicates if VSAM data sets are tracked for this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_KVSAM=Y/N

**AA\_LDATE=**

Purpose: Contains the date of the last allocation that resulted in one of the application fields being updated in yyyy/mm/dd format.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_LDATE=xxxxxxxxxx

**AA\_PERMC=**

Purpose: Indicates the total amount of space allocated on DASD for permanent data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_PERMC=*nnnnnnnnnn*

where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_PERMH=**

Purpose: Indicates the largest amount of space allocated on DASD for permanent data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_PERMH=*nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_PERMM=**

Purpose: Specifies the maximum amount of space allowed for permanent data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_PERMM=*nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_PERMP=**

Purpose: The percentage of the permanent data set budget currently being used.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_PERMP=*nnn*

where *nnn* is a whole number percentage from 0-100

**AA\_PHSM=**

Purpose: Indicates if HSM data set allocations are included as part of the permanent data set allocations.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_PHSM=*Y/N*

**AA\_PTEMP=**

Purpose: Indicates if temporary data set allocations are included as part of the permanent data set allocations.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_PTEMP=Y/N

**AA\_PVSAM=**

Purpose: Indicates if VSAM data set allocations are included as part of the permanent data set allocations.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_PVSAM=Y/N

**AA\_TEMPC=**

Purpose: Indicates the total amount of space allocated on DASD for temporary data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_TEMPC=nnnnnnnnnn

where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_TEMPH=**

Purpose: Indicates the largest amount of space allocated on DASD for temporary data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_TEMPH=nnnnnnnnnn

where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_TEMPM=**

Purpose: Specifies the maximum amount of space allowed for temporary data sets assigned to this application

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_TEMPM=nnnnnnnnnn

where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_TEMPP=**

Purpose: The percentage of the temporary data set budget currently being used.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_TEMPP=*nnn*

where *nnn* is a whole number percentage from 0-100

**AA\_UFLD*n*=**

Purpose: Application-defined user fields 1 through 3.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_UFLD*n*=*xxxxxxxxxx*

where *n* is a user field from 1-3; *xxxxxxxxxx* is the field name

AA\_UFLD1 can contain up to 8 characters

AA\_UFLD2 and 3 can contain up to 10 characters each

**AA\_UNAME=**

Purpose: Application-defined user name.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_UNAME=*xxxxxxxxxx*

where *xxxxxxxxxx* is the user name up to 20 characters

If there is a blank in the application user name, enclose the user name in quotes.

**AA\_VLCNT=**

Purpose: Indicates the number of volumes that contain at least one data set included in the application's allocation amounts.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: AA\_VLCNT=*nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AA\_VSAMC=**

Purpose: Indicates the total amount of space allocated on DASD for VSAM data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: `AA_VSAMC=nnnnnnnnnn`  
where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_VSAMH=**

Purpose: Indicates the largest amount of space allocated on DASD for VSAM data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: `AA_VSAMH=nnnnnnnnnn`  
where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_VSMM=**

Purpose: Indicates the maximum amount of space allowed for VSAM data sets assigned to this application.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: `AA_VSMM=nnnnnnnnnn`  
where *nnnnnnnnnn* is a numeric amount from 0-9223372036854775807

**AA\_VSAMP=**

Purpose: The percentage of the VSAM data set budget currently being used.

Allowed in: INC and EXC in the AUTOAPPL function

Syntax: `AA_VSAMP=nnn`  
where *nnn* is a whole number percentage from 0-100



**AA\_WTHRS=**

**Purpose:** Specifies a percentage of the budget that can be used by the application before a warning message is issued.

**Allowed in:** INC and EXC in the AUTOAPPL function

**Syntax:** AA\_WTHRS=*nnn*

where *nnn* is a whole number percentage from 0-100

**AC\_CODE=**

**Purpose:** Specifies the value in the IBM ACCODE field. Normally, this field is used in conjunction with user-generated tape labels. The tape manager for CA, CA1, uses the field to indicate various special tapes, such as Off-site, permanent hold, and so on.

**Allowed in:** INC and EXC and rule SET parameter for function SETEXPDT

**Syntax:** AC\_CODE = *xxxxxxxx*

where *xxxxxxxx* represents tape types.

**ACF2USER=**

**Purpose:** Contains the user name (CA-ACF2 system). This is a 24-byte value. For ACF2 users, this parameter should be used instead of USER or RACF.

**Note:** This parameter is not available for function FDRASIST.

**Allowed in:** INC and EXC

**Syntax:** ACF2USER=*xxxxxxxx*

where *xxxxxxxx* is a valid CA-ACF2 user name 1–24 characters long.

**ACT\_COUNT=**

**Purpose:** Optional parameter that specifies the maximum to the number of records to which actions can be taken. ACT\_EVENTID and ACT\_JOBNAME keywords identify the possible actions to take. ACT\_COUNT determines the maximum number of records those actions will be taken against. If ACT\_COUNT is not specified the action will be taken on all records in the result group for the associated SET statement, unless limited by use of the ACT\_SUM\_LIM keyword.

**Allowed in:** Rule SET parameter for AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.

**Syntax:** ACT\_COUNT=nnnnn

where nnnnn is the maximum to the number of records to which actions can be taken (1–9999)

**ACT\_EVENTID=**

**Purpose:** Specifies an event to issue for each record in the SET result group. This action is applied to the result group of records that match the SET statement INC and EXC statements. The action is applied after any SORT= specification is processed, which means the events will be generated in sort order. The number of events to issue can be limited by ACT\_SUM\_LIM or ACT\_COUNT keywords.

ACT\_EVENTID and ACT\_JOB specify the actions to take. Every SET statement should have at least one of these keywords or no action will be taken. Both statements can be used; however, care should be taken when doing this. Care must also be used when specifying ACT\_EVENTID= in an AUTOx type FLST or RLST member.

**Allowed in:** Rule SET parameter for AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.

**Syntax:** ACT\_EVENTID=xxxxx

where xxxxx is the 5-character value that corresponds to a valid event definition in parmlib member SMEVNTxx

**ACT\_JOB=**

**Purpose:** Specifies the name of a member containing skeleton JCL to be submitted using the AutoOPERATOR Skeleton Tailoring facility. The skeleton tailoring facility is documented in the AutoOPERATOR Basic Automation Guide. The member name must contain skeleton JCL and be located in BBISLIB data set of the MVS RM started task.

ACT\_JOB causes the submission of a single job, with the pool/volume/data set names passed to the Skeleton Tailoring facility through stem variables. The JCL member should make use of the )DO capability of Skeleton Tailoring to insure all stem values are processed by the submitted job. This is described in detail below.

ACT\_JOB is recommended over ACT\_EVENTID if the result of the ACT\_EVENTID will be to submit a job. Using ACT\_EVENTID in conjunction with AutoOPERATOR rules to submit a job causes a job to be submitted for each event, each record in the FLST/RLST result group (as limited by a count or sum limit). Using ACT\_JOBNAME causes all of the result records to be passed to the Skeleton Tailoring facility in a single iteration using stem variables.

ACT\_JOB and ACT\_EVENTID specify the actions to take. Every SET statement should have at least one of these keywords, or no action will be taken. Both statements can be used, however, care should be taken when doing this.

**Allowed in:** Rule SET parameter for AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.

**Syntax:** ACT\_JOB=xxxxxxx

where xxxxxxx is the 1–8 character name of the member that contains the skeleton JCL in the UBBPROC data set in the AO started task

**ACT\_SUM\_FLD=**

- Purpose:** Optional parameter that causes a running total of the specified field to be maintained for each record against which a specified action is taken. When using ACT\_SUM\_FLD, ACT\_SUM\_LIM can be used to stop action from being taken after the running total reaches a specified value.
- Allowed in:** Rule SET parameter for AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.
- Syntax:** ACT\_SUM\_FLD=xxxxxxxxxxx
- where xxxxxxxxxxxx is a 1–11-character valid INC and EXC field for the function associated with the RLST; the field name must be that of a numeric field

**ACT\_SUM\_LIM=**

- Purpose:** Optional parameter that can be used to limit the number of records to be included in any specified action. If specified, ACT\_SUM\_FLD= must also be specified. A running total of the field specified in ACT\_SUM\_FLD is maintained as each action is taken. At the completion of the action the total is updated. Prior to taking the next action, the total is compared to ACT\_SUM\_LIM. If the total is less than the limit, the next action is taken. If the total is equal to or greater than the limit, action processing is bypassed for the remainder of the records in the SET group.
- Allowed in:** Rule SET parameter for AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.
- Syntax:** ACT\_SUM\_LIM=nnnnnnnnnn
- where nnnnnnnnnn is the maximum value reached by ACT\_SUM\_FLD (0-9223372036854775807)

**AD\_ALVL1=**

- Purpose:** Contains the application name (Application Level 1).
- Allowed in:** INC and EXC in the AUTODS function
- Syntax:** AD\_ALVL1=xxxxxxxxxxxxxxxxxx

**AD\_ALVL2=**

Purpose: Contains the application name (Application Level 2).

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_ALVL2=xxxxxxxxxxxxxxxxx

**AD\_ALVL3=**

Purpose: Contains the application name (Application Level 3).

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_ALVL3=xxxxxxxxxxxxxxxxx

**AD\_ALVL4=**

Purpose: Contains the application name (Application Level 4).

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_ALVL4=xxxxxxxxxxxxxxxxx

**AD\_BLKEF=**

Purpose: Specifies the percentage of blocking efficiency.

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_BLKEF=*nnn*

where *nnn* is a whole number percentage up to 100

**AD\_BLKSZ=**

Purpose: Contains physical block size or VSAM control interval size for the data set.

For VSAM KSDS, ESDS and RRDS, the control interval size is displayed. For all other data set types, the physical block size is displayed. For VSAM data sets, the physical block size and control interval size are obtained from the catalog.

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_BLKSZ=*nnnnn*

where *nnnnn* is from 0 to 32760

### **AD\_BLKTR=**

**Purpose:** Contains the number of physical blocks that will fit on one track. The value relates to the Percent Efficiency value.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_BLKTR=nnnnnnnnnn  
where nnnnnnnnnn is a number from 0-2147483647

### **AD\_CASPL=**

**Purpose:** Contains the number of VSAM control area splits performed on the data set.  
  
This number is for the entire VSAM data set if it spans multiple volumes. Non-VSAM data sets will show NA in this field.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_CASPL=nnnnnnnnnn  
where nnnnnnnnnn is a number from 0-2147483647

### **AD\_CAT=**

**Purpose:** Contains the catalog status of the data set.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_CAT=x  
  
where x is one of the following values:  
  
C = The data set is cataloged and resides on the volume shown.  
N = The data set is not cataloged, but resides on the volume shown.  
D = The data set is not cataloged, but resides on the volume shown.  
However, there is a data set with the same name on a different volume that is cataloged.

### **AD\_CDATE=**

**Purpose:** Contains the 10-character creation date of the data set in yyyy/mm/dd format.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_CDATE=xxxxxxxxxx

**AD\_CHG=**

**Purpose:** Contains an indicator of whether the data set has been opened for output (changed).

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_CHG=Y/N

**AD\_CISPL=**

**Purpose:** Contains the number of VSAM control interval splits performed on the data set.

This number is for the entire VSAM data set if it spans multiple volumes. Non-VSAM data sets will show NA in this field.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_CISPL=nnnnnnnnnnnn

where nnnnnnnnnnn is a number from 0-2147483647

**AD\_DAYS=**

**Purpose:** Contains the number of days since the data set was opened.

This number is calculated by subtracting the last reference date from the current date. If the data set was never opened, this field will show NA.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_DAYS=nnnnnnnnnnnn

where nnnnnnnnnnn is a number from 0-2147483647

**AD\_DCLAS=**

**Purpose:** Contains the SMS-assigned data class or one of the following values for the data set:

NONE = The data set is cataloged and non-SMS managed.

DUPLIC = The data set is a duplicate (not cataloged).

UNDET = The data set's SMS status could not be determined.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_DCLAS=xxxxxxx

### **AD\_DOCC=**

Purpose: Contains the device occupancy percentage.

Allowed in: IN/EXC in the AUTODS function

Syntax: **AD\_DOCC=***nnn*

where *nnn* is a percentage between 0 and 100.

### **AD\_DSN=**

Purpose: Specifies the data set name.

Allowed in: INC and EXC in the AUTODS function

Syntax: **AD\_DSN=***xxxxxxxxxx*

where *xxxxxxxxxx* is a data set name up to 44 characters

### **AD\_DSORG=**

Purpose: Contains the data set file organization and access method used to manage the data set.

Allowed in: INC and EXC in the AUTODS function

Syntax: **AD\_DSORG=***xxxx*

where *xxxx* is one of the following values:

- PS = Physical Sequential (QSAM)
- PO = Partitioned Data Set (BPAM)
- PDSE = Partitioned Data Set Extended (LIBRARY)
- VS = VSAM
- DA = Direct Access (BDAM)
- IS = Indexed Sequential (ISAM)
- = The data set organization could not be determined or the data set was never opened.

### **AD\_EXTS=**

Purpose: Contains the number of extents occupied by the data set on the volume.

Allowed in: INC and EXC in the AUTODS function

Syntax: **AD\_EXTS=***nnnnnnnnnnnn*



where *nnnnnnnnnn* is a number from 0-2147483647

#### **AD\_GROUP=**

**Purpose:** The SMS storage group name displayed if the AUTODS function is associated with an AUTOPOOL GROUP= function.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_GROUP=*xxxxxxxxxx*

where *xxxxxxxxxx* is an SMS storage group up to 30 characters

#### **AD\_LDATE=**

**Purpose:** Contains the last date the data set was opened in yyyy/mm/dd format.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_LDATE=*xxxxxxxxxx*

#### **AD\_LRECL=**

**Purpose:** Contains the maximum record length for the data set.

For VSAM data sets, the record length is obtained from the catalog.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_LRECL=*nnnnn*

where *nnnnn* is from 0 to 99999

#### **AD\_MCLAS=**

**Purpose:** Contains the SMS-assigned management class or one of the following values:

NONE = The data set is cataloged and non-SMS managed.

DUPLIC = The data set is a duplicate (not cataloged).

UNDET = The data set's SMS status could not be determined

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_MCLAS=*xxxxxxxx*

### **AD\_POOL=**

**Purpose:** The pool name displayed if the AUTODS function is associated with an AUTOPOOL POOL= function.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_POOL=xxxxxxx  
where xxxxxxx is a pool name up to 8 characters

### **AD\_POOLT=**

**Purpose:** Contains the pool type.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_POOLT=S/P  
where *S* is for subpool and *P* is for user pool

### **AD\_PUSED=**

**Purpose:** Contains the percentage of allocation that is used.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_PUSED=nnn  
where *nnn* is a percentage up to 100 in whole numbers (no decimal)

### **AD\_REBLK=**

**Purpose:** Contains the reblockable indicator, which determines whether the data set can be reblocked by the system when being moved from one device geometry to another.

This is also known as System Determined Blocksize and typically allocates a blocksize that uses the space on the device most efficiently. N indicates that the data set is not allocated with System Determined Blocksizing and must be manually reblocked when moved from one device geometry to another.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_REBLK=Y/N

**AD\_RECFCM=**

Purpose: Contains the data set record format.

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_RECFCM=xxxxx

where xxxx is one each of the following values:

The record format indicates the type of record access along with the general format of the records and blocks. Indicator meanings are:

**Non-VSAM data sets:**

F = Fixed length blocks

V = Variable length blocks

U = Undefined block lengths

B = Records are blocked

S = Records span multiple blocks

M = Records contain machine control characters

A = Records contain ANSI printer control characters

---- = The data set organization could not be determined or the data set was never opened.

**VSAM data sets:**

ESDS = Entry-Sequenced data set

KSDS = Key-Sequenced data set

LDS = Linear data set

PAGE = System page data set

UCAT = User catalog

VVDS = ICF catalog system data set

---- = The data set organization could not be determined or the data set was never opened.

**AD\_SCLAS=**

Purpose: Contains the SMS-assigned storage class or one of the following values:

NONE = The data set is cataloged and non-SMS managed.

DUPLIC = The data set is a duplicate (not cataloged).

UNDET = The data set's SMS status could not be determined

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_SCLAS=xxxxxx

**AD\_SIZE=**

Purpose: Contains the data set size in kilobytes (one kilobyte equals 1024 bytes) on the volume

Allowed in: INC and EXC in the AUTODS function

Syntax: **AD\_SIZE=nnnnnnnnnn**  
where *nnnnnnnnnn* is a number from 0-2147483647

**AD\_SMSI=**

Purpose: Specifies the SMS status of the volume

Allowed in: INC and EXC in the AUTODS function

Syntax: **AD\_SMSI=xx**  
where xx is one of the following values:

M = SMS managed  
QA = SMS quiesced all  
QN = SMS quiesced new  
DA = SMS disabled all  
DN = SMS disabled new  
UN = Not SMS managed  
NA = Unknown

**AD\_TRKSA=**

Purpose: Contains the number of tracks allocated

Allowed in: INC and EXC in the AUTODS function

Syntax: **AD\_TRKSA=nnnnnnnnnn**  
where *nnnnnnnnnn* is a number from 0-2147483647

**AD\_TRKSF=**

**Purpose:** Contains the number of tracks unused by the data set on the volume.

For VSAM data sets, the number of tracks unused is calculated from the high-allocated RBA and high-used RBA values for each volume. These values are obtained from the catalog.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_TRKSF=nnnnnnnnnnnn

where nnnnnnnnnnn is a number from 0-2147483647

**AD\_TRKSU=**

**Purpose:** Contains the number of tracks used by the data set on the volume.

For VSAM data sets, the number of tracks used is calculated from the starting RBA and high-used RBA values for each volume. These values are obtained from the catalog.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_TRKSU=nnnnnnnnnnnn

where nnnnnnnnnnn is a number from 0-2147483647

**AD\_VOL=**

**Purpose:** Specifies the volume name.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_VOL=xxxxxx

where xxxxxx is a volume name up to 6 characters

**AD\_VOLSQ=**

**Purpose:** Specifies the volume sequence number for the data set.

**Allowed in:** INC and EXC in the AUTODS function

**Syntax:** AD\_VOLSQ=nnnnn

where nnnnn is 1 to 32767

**AD\_XDATE=**

Purpose: Specifies the expiration date for the data set in yyyy/mm/dd format.

Allowed in: INC and EXC in the AUTODS function

Syntax: AD\_XDATE=xxxxxxxxxx

**ALCTYPE=**

Purpose: Contains and specifies unit of space allocation. Applies to both primary and secondary space quantities.

Allowed in: INC and EXC and rule SET parameter for function SPACSQTY

Syntax: ALCTYPE=xxx

where xxx is a valid allocation type from the following list:

KB	A block allocation where PQTY and SQTY are assumed to be kilobytes
MB	A block allocation where PQTY and SQTY are assumed to be megabytes
TRK	Tracks
CYL	Cylinders
KAV	AVGREC=K
MAV	AVGREC=M
UAV	AVGREC=U
BLK	Blocks

Default: None

**ALTPOOL=**

Purpose: Specifies the name of an alternate pool to search for additional space during secondary space allocation when no volumes are available in the current pool. The alternate pool search uses MNTYPE=ALL (mount type). Also see the parameters SPACVOLA, MNTYPE, and USECPOOL.

Allowed in: Rule SET parameter for function SPACVOLA

Syntax: ALTPOOL=xxxxxxxx

where xxxxxxxx is a valid pool name 1–8 characters long

Default: None

**AP\_CTIGC=**

Purpose: Specifies the largest contiguous free cylinders.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_CTIGC=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is from 0-2147483647

**AP\_CTIGT=**

Purpose: Specifies the largest contiguous free tracks.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_CTIGT=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_FREEC=**

Purpose: Specifies the total count of free cylinders.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_FREEC=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_FREED=**

Purpose: Specifies the total number of free data set control blocks (DSCB) for all volumes.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_FREED=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_FREET=**

Purpose: Specifies the total count of free tracks.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_FREET=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_FREEV=**

Purpose: Specifies the total number of free index records in a volume's VTOC index.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_FREEV=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_FREEEX=**

Purpose: Specifies the total number of free extents.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_FREEEX=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_FSIZE=**

Purpose: Specifies the free size (in MB) of unused space in the pool.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_FSIZE=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647



**AP\_GROUP=**

Purpose: Specifies the pool, group, or subpool name.

Allowed in: INC/EXC in the AUTOPOOL function

Syntax: `AP_GROUP=xxxxxxxx`

where *xxxxxxxx* is a 1-30 character pool, group, or subpool name

**AP\_HDFUL=**

Purpose: Specifies the VVDS percentage full high-water mark.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_HDFUL=nnn`

where *nnn* is a percentage from 1-100

**AP\_HFULL=**

Purpose: Specifies the high-water mark volume percentage full, which is the percentage full of the highest utilized volume in the pool.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_HFULL=nnn`

where *nnn* is percentage up to 100

**AP\_HIFUL=**

Purpose: Specifies the VTOC percentage full high-water mark.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_HIFUL=nnn`

where *nnn* is a percentage from 1-100

**AP\_HREEC=**

Purpose: Specifies the high-water mark count of free cylinders.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: **AP\_HREEVC=***nnnnnnnnnn*  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_HREED=**

Purpose: Specifies the high-water mark count of free DSCBs.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: **AP\_HREED=***nnnnnnnnnn*  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_HREET=**

Purpose: Specifies the high-water mark count of free tracks.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: **AP\_HREET=***nnnnnnnnnn*  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_HREEV=**

Purpose: Specifies the high-water mark count of free VIR

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: **AP\_HREEV=***nnnnnnnnnn*

**AP\_HREEX=**

Purpose: Specifies the high-water mark count of free extent

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: **AP\_HREEX=***nnnnnnnnnn*  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_HVFRG=**

Purpose: Specifies the high-water mark fragmentation index.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_HVFRG=nnn`  
where *nnn* is percentage up to 100

**AP\_HVFUL=**

Purpose: Specifies the high-water mark VTOC percentage full.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_HVFUL=nnn`  
where *nnn* is percentage up to 100

**AP\_LDFUL=**

Purpose: Specifies the VVDS percentage full low-water mark.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_LDFUL=nnn`  
where *nnn* is a percentage from 1-100

**AP\_LFULL=**

Purpose: Specifies the low-water mark volume percentage full.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_LFULL=nnn`  
where *nnn* is percentage up to 100

**AP\_LIFUL=**

Purpose: Specifies the VTOC index percentage full low-water mark.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_LIFUL=nnn*

where *nnn* is a percentage from 1-100

**AP\_LPRIC=**

Purpose: Specifies the largest primary allocation (cylinders).

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_LPRIC=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_LPRIT=**

Purpose: Specifies the largest primary allocation (tracks).

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_LPRIT=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_LREEC=**

Purpose: Specifies the low-water mark free count of cylinders.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_LREEC=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_LREED=**

Purpose: Specifies the low-water mark free count of DSCBs.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_LREED=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_LREET=**

Purpose: Specifies the low-water mark free count of tracks.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_LREET=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_LREEV=**

Purpose: Specifies the low-water mark free count of VIR.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `LP_LREEV=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_LREEX=**

Purpose: Specifies the low-water mark free count of extent.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: `AP_LREEX=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number from 0-2147483647

**AP\_LVFRG=**

Purpose: Specifies the low-water mark fragmentation index.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_LVFRG=nnn*  
where *nnn* is percentage up to 100

**AP\_LVFUL=**

Purpose: Specifies the low-water mark of VTOC percentage full.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_LVFUL=nnn*  
where *nnn* is percentage up to 100

**AP\_PERFL=**

Purpose: Specifies the pool percentage full based on used space as it relates to total space.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_PERFL=nnn*  
where *nnn* is percentage up to 100

**AP\_POOL=**

Purpose: Specifies the pool, group, or subpool name.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_POOL=xxxxxxxxxx*  
where *xxxxxxxxxx* is a valid pool, group, or subpool name up to 30 characters

**AP\_TSIZE=**

Purpose: Specifies the total size (in MB) of space in the pool.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_TSIZE=nnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_TYPE=**

Purpose: Specifies the pool type.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_TYPE=xxxxxxxx*

Valid values are: POOL, SMSPOOL, and GROUP

**AP\_USIZE=**

Purpose: Specifies the amount of allocated space in the pool (used size).

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_USIZE=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_VOLC=**

Purpose: Specifies the number of online volumes in this pool on the collecting OS/390 system.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_VOLC=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AP\_VOLD=**

Purpose: Specifies the volume drop count (due to errors).

Number of volumes in the pool that were not collected due to collection errors. The totals for the pool might be invalid due to these volumes not being included. The SVOS joblog will contain error messages indicating the errors encountered.

Allowed in: INC and EXC in the AUTOPOOL function

Syntax: *AP\_VOLD=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AUTOLEV=**

Purpose: Contains an 8-character literal AUTOLEV $x$ , where  $x$  is a number indicating the current automation level for the resource being automated

AUTOLEV0 indicates that multiple levels are not being used. AUTOLEV1 indicates the first level of a multiple level request. For multiple level automation this field must be used either in this FLST/RLST member or contained within event text of any event issued and then referenced in the AutoOPERATOR rule that looks for the event.

Allowed in: INC and EXC in the AUTO functions

Syntax: AUTOLEV=xxxxxxxx

**AV\_CTIGC=**

Purpose: Contains the largest single extent in full cylinders available for allocation.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_CTIGC=nnnnnnnnnnnn

where nnnnnnnnnnn is a number from 0-2147483647

**AV\_CTIGT=**

Purpose: Contains the largest single extent in tracks available for allocation.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_CTIGT=nnnnnnnnnnnn

where nnnnnnnnnnn is a number from 0-2147483647

**AV\_DEV=**

Purpose: Contains the generic unit name for the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_DEV=xxxxxxxx

where xxxxxxxx is a 1-8 character generic unit name



**AV\_FRAGI=**

Purpose: Contains the fragmentation index value of the volume. The higher the value, the more fragmented the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_FRAGI=nnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_FREEC=**

Purpose: Contains the number of free cylinders on the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_FREEC=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_FREED=**

Purpose: Contains the number of free (Format 0) DSCBs on the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_FREED=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_FREET=**

Purpose: Contains the number of free tracks on the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_FREET=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_FREEV=**

Purpose: Contains the number of free VIRs (VTOC index records) on the volume

Allowed in: INC and EXC in the AUTOVOL function.

Syntax: *AV\_FREEV=nnnnnnnnnnnn*  
where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_FREEX=**

Purpose: The total amount of free extents on the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_FREEX=nnnnnnnnnnnn*  
where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_FSIZE=**

Purpose: The number of tracks not used on the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_FSIZE=nnnnnnnnnnnn*  
where *nnnnnnnnnnnn* is a number of tracks from 0-2147483647

**AV\_FULL=**

Purpose: Contains the percentage of used space to total space for the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_FULL=nnn*  
where *nnn* is a percentage up to 100  
This is a whole number, no decimal places.

**AV\_IPCTF=**

Purpose: Specifies the VTOC index percentage full.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_IPCTF=*nnn*

where *nnn* is a percentage from 0-100

**AV\_ISIZE=**

Purpose: Specifies the total size of the VTOC index.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_ISIZE=*nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_LPRIC=**

Purpose: Contains the largest possible primary extent in cylinders.

This is the sum of the 5 largest extents on the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_LPRIC=*nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_LPRIT=**

Purpose: Contains the largest possible primary extent in tracks.

This is the sum of the 5 largest extents on the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_LPRIT=*nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

**AV\_MNT=**

Purpose: Contains an indicator of how the volume is mounted.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_MNT=*xxx*

where *xxx* is one of the following indicators:

PUB Public  
PVT Private  
STG Storage  
SYS System

**AV\_PCNT=**

Purpose: Contains the pool count.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_PCNT=*n*

where *n* is a pool number 1-8

**AV\_PID=**

Purpose: Contains the physical disk ID.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_PID=*xxxxxxxxxxx*

where *xxxxxxxxxxx* is a physical disk name of 1-11 characters

**AV\_POOL*n*=**

Purpose: Specifies the pool name for pool number 1-8

Allowed in: INC and EXC in the AUTOVOL function

Syntax: AV\_POOL*n*=*xxxxxxx*

where *n* is the pool number (1-8) and *xxxxxxx* is the pool name

**AV\_PTYP*n*=**

Purpose: Specifies the pool type.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: `AV_PTyp $n=x$`

where  $n$  is the pool number (1-8) and  $x$  is S for subpool or P for user pool

**AV\_SMSGP=**

Purpose: Contains the SMS-assigned storage group name.

If the volume is not SMS managed, this field will be blank.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: `AV_SMSGP=xxxxxxxxxxxxxxxx`

where `xxxxxxxxxxxxxxxx` is a 1-30 character SMS group name

**AV\_SMSI=**

Purpose: Contains the SMS status of the volume.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: `AV_SMSI=xx`

where `xx` is one of the following values:

- M = SMS managed
- QA = SMS quiesced all
- QN = SMS quiesced new
- DA = SMS disabled all
- DN = SMS disabled new
- UN = Not SMS managed
- NA = Unknown

**AV\_TSIZE=**

Purpose: The total volume size in tracks.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: `AV_TSIZE=nnnnnnnnnn`

where `nnnnnnnnnn` is a number of tracks from 0-2147483647

**AV\_UCB=**

Purpose: Contains the device number for the volume (4 hexadecimal digits). Masking characters are not allowed with this parameter; it is treated as a number.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_UCB=hhhh*  
where *hhhh* is a hexadecimal number from 0001 to FFFF

**AV\_USIZE=**

Purpose: The amount of space used on the volume in tracks.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_USIZE=nnnnnnnnnn*  
where *nnnnnnnnnn* is a number of tracks from 0-2147483647

**AV\_VOL=**

Purpose: Contains the volume serial number.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_VOL=xxxxxx*  
where *xxxxxx* is a 1-6 character volume serial number

**AV\_VPCTF=**

Purpose: Specifies the VVDSF percentage full.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_VPTCF=nnn*  
where *nnn* is a percentage from 0-100

**AV\_VSIZE=**

Purpose: Specifies the total size of the VVDS.

Allowed in: INC and EXC in the AUTOVOL function

Syntax: *AV\_VSIZE=nnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

#### **AV\_VTOCF=**

**Purpose:** Contains the percentage of used VTOC space.

**Allowed in:** INC and EXC in the AUTOVOL function

**Syntax:** *AV\_VTOCF=nnn*

where *nnn* is a percentage up to 100

This percentage is a whole number, no decimal places.

#### **AV\_VTOCI=**

**Purpose:** Contains the VTOC index status.

**Allowed in:** INC and EXC in the AUTOVOL function

**Syntax:** *AV\_VTOCI=xxx*

where *xxx* is one of the following values:

ACT = VTOC Index is defined and active

INA = VTOC Index is defined but is not active

UND = VTOC Index is not defined

DIS = VTOC has had an index that has been disabled

#### **AV\_VTOCZ=**

**Purpose:** Contains the volumes VTOC size in tracks.

**Allowed in:** INC and EXC in the AUTOVOL function

**Syntax:** *AV\_VTOCZ=nnnnnnnnnnnn*

where *nnnnnnnnnnnn* is a number from 0-2147483647

### **AVL=**

**Purpose:** Specifies the average block size for BLK allocations and record length for KAV, MAV, and UAV allocations.

**Allowed in:** Rule SET parameter for function SPACSQTY

**Syntax:** AVL=*nnnnn*

where *nnnnn* is a number in the range 1–32767. Default is 9000.

### **BACKCMD=**

**Purpose:** Specifies whether the DFHSM backup initiated by command (instead of automatically) is allowed or disallowed. If BACKCMD=Y is specified, HSMBACKP is activated both for backups initiated automatically and on command. If BACKCMD=N, HSMBACKP is activated only for automatic backup processing.

**Allowed in:** Rule SET parameter for function HSMBACKP

**Syntax:** BACKCMD= Y/N

**Default:** BACKCMD=Y

### **BACKUP=**

**Purpose:** For MAINVIEW SRM function HSMBACKP, specifies the inclusion of data sets and volumes in DFHSM backup processing. BACKUP=N excludes selected resources from DFHSM backup processing. The default is NO.

For MAINVIEW SRM function HSMmigRT, specifies that migration direct from ML0 to ML2 is allowed without a DFHSM backup copy of the data set. The default is Y.

**Allowed in:** Rule SET parameter for functions HSMBACKP, HSMmigRT

**Syntax:** BACKUP= Y/N



**BLKSIZE=**

**Purpose:** Specifies or contains the block size of a data set. A specification of zero requests a system determined blocksize (if supported by your release of OS/390 and DFP).

In a filter list specification, greater than (>) or less than (<) signs can be used instead of the equals sign (=).

**Allowed in:** INC and EXC, and rule SET parameter for function OPTBLKSZ

**Syntax:** BLKSIZE=<>*nnnnn*

where *nnnnn* is a number in the range 0–32,760.

**BUFSP=**

**Purpose:** Contains and specifies the buffer space for the cluster or the data component of VSAM data sets. Any existing buffer space specification is overridden.

**Note:** A BUFSP value that is too small will be ignored by VSAM. Generally, a value less than {2 x data CFSIZE} for non-indexed files or a value less than {2x data CFSIZE+1x index CFSIZE} for indexed files will be too small.

**Allowed in:** INC and EXC and rule SET parameter for function VSAMCNTL

**Syntax:** BUFSP=*nnnnnn*

where *nnnnnn* is a number in the range 0–16776704.

**CAL=**

**Purpose:** Specifies that dates must be adjusted by the specifications in the MAINVIEW SRM non-working day calendar; for example, an expiration date is calculated using the retention period in days plus the number of non-working days in that period.

**Allowed in:** Rule SET parameter for functions HSMMCCNV, HSMMIGRT, and SETEXPDT

**Syntax:** CAL=Y/N

**Default:** None

**CALAGE=**

**Purpose:** Contains the calendar-adjusted unreferenced day count set by the HSMmigrt function. The number of non-working days is subtracted from the unreferenced day count.

**Allowed in:** INC and EXC

**Syntax:** CALAGE=nnnn

where *nnnn* is a number in the range 0–9999.

**CANDIDATE=**

**Purpose:** Specifies whether the IDCAMS DEFINE VOL=( ) list for a VSAM data set is used as an indicator of secondary volume allocation. If CANDIDATE=Y, the *number* of DEFINED volumes is used to select that many candidate volumes from the MAINVIEW SRM pool. If CANDIDATE=N, no secondary volumes are assigned or available through normal OS/390 processing (but SPACVOLA can be used to assign secondary volumes from a MAINVIEW SRM pool). Note that with specification of the CANDIDATE parameter, the candidate volumes are assigned from the pool, not from those volumes specified in the IDCAMS DEFINE VOL list. Note also that CANDIDATE overrides NVOLVSAM.

**Allowed in:** Rule SET parameter for function DASDPOOL

**Syntax:** CANDIDATE=Y/N

**Default:** CANDIDATE=N

**CAT=**

**Purpose:** Contains the name of the catalog for a data set. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** CAT=xxxxxxxx

where *xxxxxxxx* is a valid catalog name 1–44 characters long.

**CATALOG=**

**Purpose:** Contains and specifies the removal of the CATALOG parameter during VSAM cluster definition.

**Allowed in:** INC and EXC and rule SET parameter for function VSAMCNTL

**Syntax:** CATALOG= Y/N/BLANK

If Y is specified, the CATALOG parameter is allowed in the DEFINE CLUSTER definition; if N is specified, the CATALOG parameter is removed from the DEFINE CLUSTER definition when used in an INC and EXC statement. Blank indicates that no specification for CATALOG was made in the IDCAMS statements.

**Note:** When used as a selection parameter, this parameter is only valid for VSAM data sets.

### CISIZE=

**Purpose:** Specifies the value of the data or cluster and/or index control interval size in the corresponding component of the IDCAMS control cards. Any existing specification of control interval size is overridden.

**Allowed in:** INC and EXC and rule SET parameter for function VSAMCNTL

**Syntax:** CISIZE=nnnnnn or CISIZE=(nnnnnn,nnnnnn)

where the first nnnnnn is the data or cluster control interval size and the second nnnnnn is the index control interval size (0–999999).

### COMP=

**Purpose:** Specifies whether a tape data set is to be compressed. (IDRC must be supported by the allocated cartridge device.)

**Allowed in:** Rule SET parameter for function TAPECOMP

**Syntax:** COMP= Y/N

### CONTIG=

**Purpose:** Contains and specifies whether a data set is allocated with contiguous space required.

**Allowed in:** INC and EXC and rule SET parameter for function SPACSQTY

**Syntax:** CONTIG= Y/N

**CRITBIAS=**

**Purpose:** Defines the number of data sets that can reside on the volume before the current allocation.

**Allowed in:** Rule SET parameter for function DASDPOOL, FDRASIST, and SMSSELECT

**Syntax:** CRITBIAS=*n*  
where *n* is a number from 1 to 9

**Default:** None

**Note:** This parameter works only in conjunction with VOLSEL=CRITDSN

**CRITEMC=**

**Purpose:** Specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes. If yes, data sets specified with CRITDSN should not reside on the same EMC Physical volume nor the same MVS Logical volume. If no, data sets specified with CRITDSN can reside on the same EMC physical volume. Default is NO.

**Allowed in:** Rule SET parameter for function DASDPOOL, FDRASIST, and SMSSELECT

**Syntax:** CRITEMC=Y/N

**CRITFAIL=**

**Purpose:** Defines the allocation process if a volume meeting the criteria cannot be found. If Y, allocation fails; if N, allocation is made to the best available volume. Like USELIM, if an acceptable volume cannot be found in the first pool, the best volume from the first pool is saved and the system tries the next pools. If no acceptable volume is found, CRITFAIL is processed on the saved volume.

**Allowed in:** Rule SET parameter for function DASDPOOL, FDRASIST, and SMSSELECT

**Syntax:** CRITFAIL=Y/N

**Default:** CRITFAIL=N

**Note:** This parameter works only in conjunction with VOLSEL=CRITDSN

**CRITLIST=**

Purpose:	The table that contains allocation volumes.
Allowed in:	Rule SET parameter for function DASDPOOL, FDRASIST, and SMSSELECT
Syntax:	CRITLIST=xxxxxxx
Default:	None

**Note:** This parameter works only in conjunction with VOLSEL=CRITDSN

**Warning!** CRITDSN is resource intensive and should be used only for a *small* list of critical data sets. It should not be used without considering the impact on the system.

**CURDAY=**

Purpose:	Contains the current day of the week.
Allowed in:	INC and EXC
Syntax:	CURDAY= xxxxxxxx  where xxxxxxxx is a weekday name.

**CURSPACE=**

Purpose:	Contains the number of bytes that the data set will use after adding the current space request.
Allowed in:	INC and EXC
Syntax:	CURSPACE=nnnnnnK,M,G,T

**CURTIME=**

Purpose:	Contains the current time of day. The time is in the form of HH:MM:SS.  Allowed in: NC/EXC
Syntax:	CURTIME=nn:nn:nn  where nn:nn:nn is the time of day.

**DADSM\_FUNC=**

Purpose: Contains the current point in allocation for most EasyPOOL functions.

Allowed in: INC and EXC

Syntax: DADSM\_FUNC=xxxxxxxx,xxxxxxxx,...

where xxxxxxxx is one or more of the valid options listed below:

**JCL**

indicates that the function is processed at JFCB housekeeping

**ALLOCATE**

indicates that the function is being processed at IGGPRE00

**ALLOCATE**

This is valid for SMSSELECT if SMS\_ALLOC has been set to Y and for FDRASIST.

**EXTENDNV**

indicates that the function is being processed at IGGPRE00

EXTENDNV (extend to a new volume)

This is valid for SMSSELECT if SMS\_EXTEND has been set to Y.

**RENAME**

indicates that the function is being processed at IGGPRE00

**RENAME**

This is valid for DASDPOOL if DP\_RENAME has been set to Y and for SMSMCREN.

**DATACLAS=**

Purpose: Specifies or contains the name of a DFSMS data class. MAINVIEW SRM name masking can be used for filter list entries. Rule list entries must specify a valid data class name.

Allowed in: INC and EXC and rule SET parameter for function SMSACSDC

Syntax: DATACLAS=xxxxxxxx

where xxxxxxxx is a valid data class name 1–8 characters long.

**DD=**

**Purpose:** Contains the data definition statement name from a JCL statement. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** DD=xxxxxxx

where xxxxxx is a valid DD name 1–8 characters long.

**DEFUNIT=**

**Purpose:** Specifies the default unit name for volumes located outside STK silos.

**Allowed in:** Rule SET parameter for function STKSUPP.

**Syntax:** DEFUNIT=xxxxxxx

where xxxxxx specifies a generic unit name (3480 is the default).

**DEVTYPE=**

**Purpose:** Contains the type of device. For all functions except OPTBLKSZ, valid values are DASD, TAPE, or UNKN. For OPTBLKSZ, valid values are TAPE, 3380, and 3390.

The UNKN (unknown) device type is used for devices that are neither DASD nor TAPE and for devices that are requested by specific unit address.

**Allowed in:** INC and EXC

**Syntax:** DEVTYPE=xxxx

where xxxx is a valid device type name from the following list: DASD, TAPE, UNKN, 3380, 3390.

**DIR=**

**Purpose:** Contains and specifies the number of directory blocks for partitioned data sets.

**Allowed in:** INC and EXC and rule SET parameter for function SPACSQTY

**Syntax:** DIR=nnnn

where nnnn is a number in the range 1–9999.

**DISPn=**

**Purpose:** Contains a data set disposition. DISP1 is the first JCL DISP subparameter (status); DISP2 is the second JCL DISP subparameter (normal termination disposition); and DISP3 is the third JCL DISP subparameter (abnormal termination disposition).

**Allowed in:** INC and EXC

**Syntax:** DISPn=xxxxxx

where xxxxxx is a valid disposition from the following list:

DISP1 NEW, OLD, SHR, MOD

DISP2 DELET, KEEP, PASS, CATLG, UNCAT

DISP3 DELET, KEEP, CATLG, UNCAT

**DPORDEF=**

**Purpose:** Specifies the default response time for device selection based on DASD performance. The default value is used when response time information has not been accumulated for a device, such as when it has just been brought online.

DPORDEF can only be specified in conjunction with DPORMIN/MAX.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

**Allowed in:** Rule SET parameter for function DASDPOOL and SMSSELECT

**Syntax:** DPORDEF=nnn

where nnn specifies the default response time in milliseconds that will be substituted for unavailable information for a specific device (0-999).



**DPORMAX=**

**Purpose:** Specifies the maximum response time objective for device selection based on DASD performance.

DPORMAX is normally specified in conjunction with DPORMIN.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

**Allowed in:** Rule SET parameter for function DASDPOOL and SMSSELECT

**Syntax:** DPORMAX=*nnn*

where *nnn* specifies the maximum response time in milliseconds that will be considered for device selection based on performance (0-999).

**DPORMIN=**

**Purpose:** Specifies the minimum response time objective for device selection based on DASD performance.

DPORMIN is normally specified in conjunction with DPORMAX.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

**Allowed in:** Rule SET parameter for function DASDPOOL and SMSSELECT

**Syntax:** DPORMIN=*nnn*

where *nnn* specifies the minimum response time in milliseconds that will be considered for device selection based on performance (0-999).

**DPORSEP=**

**Purpose:** Specifies the response time value that is used to force selection of different volumes for data sets in the same jobstep. When multiple data sets are allocated in a single jobstep and device selection based on DASD performance is specified for some or all, this separation factor is added to the current response time for previously used volumes in order to increase the likelihood that new allocations will go to a different volume.

DPORSEP can only be specified in conjunction with PORMIN/MAX.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

**Allowed in:** Rule SET parameter for function DASDPOOL and SMSSELECT

**Syntax:** DPORSEP=*nnn*

where *nnn* specifies the response time in milliseconds that will be used to enhance data set separation across volumes (0-999).

**DPOWIND=**

**Purpose:** Specifies the window (length of the performance interval) that is used to analyze the response characteristics of DASD devices for selection based on performance.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

**Allowed in:** Rule SET parameter for function DASDPOOL and SMSSELECT

**Syntax:** DPOWIND=*nnnn*

where *nnnn* specifies the number of seconds over which the performance analysis will be made (0-999).

**DSN=**

**Purpose:** Contains the data set name. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** DSN=*xxxxxxxxxxxxxxxxxxxxxxxx....*

where *xxxxxxxx* is a valid data set name 1–44 characters long.

**DSNAME=**

Purpose: Synonym. See DSN.

**DSNn=**

Purpose: Specifies or contains a data set name qualifier. MAINVIEW SRM name masking can be used for filter list entries.

Allowed in: INC and EXC and rule SET parameter for function DSNCHECK

Syntax: `DSN $n$ =xxxxxxx`

where  $n$  is a number from 1–8 specifying the qualifier position in the data set name, and xxxxxx is a valid data set qualifier name 1–8 characters long.

**DSNTYPE=**

Purpose: Contains the data set name type (but only if it is specified in the JCL).

Allowed in: INC and EXC.

Syntax: `DSNTYPE=xxx`

where xxx is a value from the following list:

PDS	Partitioned data set
LIB	Extended partitioned data set
HFS	Hierarchical file system (Open OS/390)
PIP	Pipe (Open OS/390)
STR	Striped
DB2	IBM DB2 database
IAM	VSAM replacement access method. Cannot be set during NOCATLG2, SPACPRIM, or SPACSWIR processing. The required data set is not yet open when these functions are processed.

**Note:** The application collector supports only LIB, HFS, and STR.

**DSORG=**

Purpose: Contains the data set organization.

Allowed in: INC and EXC

Syntax: DSORG=xxxx

where xxxx is a value from the following list:

PS = Physical Sequential (QSAM)

PO = Partitioned Data Set (BPAM)

PDSE = Partitioned Data Set Extended (LIBRARY)

VS = VSAM

DA = Direct Access (BDAM)

IS = Indexed Sequential (ISAM)

– = The data set organization could not be determined or the data set was never opened.

**DSTYPE=**

Purpose: Contains the data set type.

Allowed in: INC and EXC

Syntax: DSTYPE=xxxx

where xxxx is a value from the following list:

PERM Permanent data set

TEMP Temporary data set

GDG Generation data set

**DYNALLOC=**

Purpose: Contains the allocation type (dynamic or JCL).

Allowed in: INC and EXC

Syntax: DYNALLOC=Y/N

**ENVIR=**

**Purpose:** Specifies the DFSMS allocation environment.

**Note:** This parameter is not available for functions SPACPRIM, SPACSECA, SPACSECB, SPACSECI, SPACSECR, SPACSWIR, and SPACVOLA.

**Allowed in:** INC and EXC

**Syntax:** ENVIR=xxxxx

where xxxxx is a value from the following list:

ALLOC	New data set allocations
RECALL	Data set recall operations
RECOVER	Data set recover operations
CONVERT	Data set convert-in-place operations
STORE	OSREQ object store environment
CHANGE	OSREQ object change environment
CTRANS	OSMC object class transition environment
other	Set by installation exit

During a rename operation, the DSNCHECK function sets this parameter to RENAME to allow different naming standards on data set renames, if desired.

**ERASE=**

**Purpose:** Contains and specifies the removal of the ERASE parameter during cluster definition.

**Allowed in:** INC and EXC and rule SET parameter for function VSAMCNTL

**Syntax:** ERASE=Y/N

If Y is specified the ERASE parameter is forced in the DEFINE CLUSTER definition; if N is specified, the ERASE parameter is removed from the DEFINE CLUSTER definition.

**Note:** When used as a selection parameter, this parameter is only valid for VSAM data sets.

## EVENTID=

Purpose:	Specifies the identifier assigned to the user event in SMEVNT $_{xx}$ .
Allowed in:	Rule SET parameter allowed in FLST or RLST of any function except USERVARS and the AUTO functions (see ACT_EVENTID= for use in AUTO functions)
	<b>Note:</b> If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.
Syntax:	EVENTID= $xxxxx$  where $xxxxx$ is the 5-character string specified on the EVNTID parameter in SMEVNT $_{xx}$ .
Required:	No
Default:	None

## EXPDT=

Purpose:	Contains and specifies the expiration date for a data set.
Allowed in:	INC and EXC and rule SET parameter for function SETEXPDT
Syntax:	EXPDT= $nnnnnn$  where $nnnnnn$ is a Julian date in the format $yyddd$ to represent a date from 1900 to 1999.  or  EXPDT= $nnnnnnnn$  where $nnnnnnnn$ is a Julian date in the format $yyyyddd$ to represent a date from any year.

## EXTENT=

Purpose:	Contains the number of extents for a data set.
Allowed in:	INC and EXC
Syntax:	EXTENT= $<>nnn$  where $nnn$ is a number in the range 1–123. The comparison operator symbol can be equals (=), greater than (>), or less than (<).

**FILESEQ=**

Purpose: Contains the file sequence number of the data set.

**Note:** This parameter is not available for functions SPACPRIM, SPACSECA, SPACSECB, SPACSECI, SPACSECR, SPACSWIR, and SPACVOLA.

Allowed in: INC and EXC

Syntax: FILESEQ=nnnnnn

where *nnnnnn* is a number in the range of 0–999999.

**FORCE=**

Purpose: Specifies whether the program specified blocksize should be overridden.

Allowed in: Rule SET parameter for functions EasyPOOL and StopX37/II

Syntax: FORCE=Y/N

If Y, the value specified will be used to override a program-specified blocksize; if N, a program specified blocksize will *not* be overridden.

Default: FORCE=N

**FUNCTION=**

Purpose: Specifies the name of the current function.

Allowed in: Filter list INC and EXC

Syntax: FUNCTION=xxxxxxxx

where *xxxxxxxx* is the eight-character function name.

**GDGVER=**

**Purpose:** Contains the relative version number relative to the current generation of a GDG data set. This parameter is valid for the HSM MIGRT and HSM DELETE functions only.

**Allowed in:** INC and EXC (set only for HSM function)

**Syntax:** GDGVER=<>*nnn*

where *nnn* is a number in the range 0–255. The comparison operator symbol can be equals (=), greater than (>), or less than (<).

**HDPORDEF=**

**Purpose:** Specifies the default response time for device selection based on DASD performance. The default value is used when response time information has not been accumulated for a device, such as when it has just been brought online.

HDPORDEF is useful only when specified in conjunction with HDPORMIN/MAX. This parameter is available only when VOLSEL=HISTDPO.

**Allowed in:** INC and EXC

**Syntax:** HDPORDEF=*nnn*

where *nnn* specifies the default response time in milliseconds that will be substituted for unavailable information for a specific device (0-999).

**Default:** HDPORDEF=50

**HDPORMAX=**

**Purpose:** Specifies the maximum response time objective for device selection based on DASD performance.

HDPORMAX is normally specified in conjunction with HDPORMIN. This parameter is available only when VOLSEL=HISTDPO.

**Allowed in:** INC and EXC

**Syntax:** HDPORMAX=*nnn*

where *nnn* specifies the maximum response time in milliseconds that will be considered for device selection based on performance (0-999).



Default: None

**HDPORMIN=**

Purpose: Specifies the minimum response time objective for device selection based on DASD performance.

HDPORMIN is normally specified in conjunction with HDPORMAX. This parameter is available only when VOLSEL=HISTDPO.

Allowed in: INC and EXC

Syntax: HDPORMIN=*nnn*

where *nn* specifies the minimum response time in milliseconds that will be considered for device selection based on performance (0-999).

Default: None

**HDPORSEP=**

Purpose: Specifies the response time value that is used to force selection of different volumes for data sets in the same jobstep. When multiple data sets are allocated in a single jobstep and device selection based on DASD performance is specified for some or all, this separation factor is added to the current response time for previously used volumes in order to increase the likelihood that new allocations will go to a different volume. This parameter is available only when VOLSEL=HISTDPO.

Allowed in: INC and EXC

Syntax: HDPORSEP=*nnn*

where *nnn* specifies the response time in milliseconds that will be used to enhance data set separation across volumes (0-999).

Default: HDPORSEP=10

**HDPOSTIM=**

**Purpose:** Specifies the starting time (each day) for a range of snapshots to be analyzed. This allows analysis of data from the same time range, or shift, each day. This parameter is available only when VOLSEL=HISTDPO.

**Allowed in:** INC and EXC

**Syntax:** HDPOSTIM=*nnnn*  
where *nnnn* specifies a time in 24-hour format (0-2359).

**Default:** HDPOSTIM=0000

**HDPOETIM=**

**Purpose:** Specifies the ending time (each day) for a range of snapshots to be analyzed. This allows analysis of data from the same time range, or shift, each day. This parameter is available only when VOLSEL=HISTDPO.

**Allowed in:** INC and EXC

**Syntax:** HDPOETIM=*nnnn*  
where *nnnn* specifies a time in 24-hour format (0-2359).

**Default:** HDPOETIM=2359

**HDPODAYS=**

**Purpose:** Specifies the days of the week to be used in determining performance statistics for pooling. This parameter is available only when VOLSEL=HISTDPO.

**Allowed in:** INC and EXC

**Syntax:** HDPODAYS=(*MO, TU, WE, TH, FR, SA, SU*)

**Default:** HDPODAYS=(*MO, TU, WE, TH, FR, SA, SU*)

**HLQ=**

**Purpose:** Contains the high-level qualifier of a data set name. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** HLQ=xxxxxxx

where xxxxxxx is a valid data set qualifier 1–8 characters long.

**HSM=**

**Purpose:** Flags a DFHSM migration or backup data set. The HSM data set flag is set only if the HSMTRACK (Y) system option has been specified.

**Allowed in:** INC and EXC (only for application collector)

SET (only for HSM functions)

**Syntax:** HSM=Y/N

**HSMDSN=**

**Purpose:** Specifies the DFHSM migration or backup data set name.

**Allowed in:** SET (only for HSM functions)

**Syntax:** HSMDSN=xxxxxxx

where xxxxxxx is a data set name 1–44 characters long.

**IMBED=**

**Purpose:** Contains and specifies the removal of the IMBED parameter during cluster definition.

**Allowed in:** INC and EXC and rule SET parameter for function VSAMCNTL

**Syntax:** IMBED=Y/N

If Y is specified the IMBED parameter is forced in the DEFINE CLUSTER definition; if N is specified, the IMBED parameter is removed from the DEFINE CLUSTER definition.

**Note:** When used as a selection parameter, this parameter is only valid for VSAM data sets.

### **JOB=**

**Purpose:** Contains the jobname (batch job, started task, TSO user). MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** JOB=xxxxxxx  
where xxxxxxx is a valid job name 1–8 characters long.

### **JOBACCTn=**

**Purpose:** Contains the *n*th field of the job card ACCT field. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** JOBACCTn=xxxxxxxxxxx  
where *n* is a number in the range of 1–3 and xxxxxxxxxxx is a character string 1–20 characters long.

### **JOBCLASS=**

**Purpose:** Contains the value of the class field of the job card of the currently executing job. MAINVIEW SRM name masking can be used.

**Note:** This parameter is not available for functions DSNCHECK, SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, or SMSACSTE.

**Allowed in:** INC and EXC

**Syntax:** JOBCLASS=x  
where *x* is a single job class character.

### **JOBSDAY=**

**Purpose:** Contains the day of the week the job was started.

**Allowed in:** INC and EXC

**Syntax:** JOBSDAY= xxxxxxx  
where xxxxxxx is the day of the week the job was initiated.

**JOBSTIME=**

Purpose: Contains the start time for the job stated in the form of HH:MM:SS.

Allowed in: INC and EXC

Syntax: `JOBSTIME=nn:nn:nn`

where *nn:nn:nn* is the hour, minute, and second when the job was started.

**JOBTYPE=**

Purpose: Specifies the type of job that issued a request

Allowed in: INC and EXC

Syntax: `HSM=STC/TSO/JOB`

where STC is a started task, TSU is a timesharing user, and JOB is a batch job.

**LABELTYP=**

Purpose: Contains the value of the LABEL parameter of the DD statement.

**Note:** This parameter is not available for functions SPACPRIM, SPACSECA, SPACSECB, SPACSECI, SPACSECR, SPACSWIR, and SPACVOLA.

Allowed in: INC and EXC

Syntax: `LABELTYP=xxx`

where *xxx* identifies the value of the LABEL parameter. Valid values are

SL	Standard labels
SUL	Standard and user labels
AL	ANSI labels
AUL	ANSI and user labels
NSL	Non-standard labels
NL	No labels
BLP	Bypass label processing
LTM	Leading tapemark

### **LEVEL=**

Purpose: Specifies the account level being processed by application.

Allowed in: INC and EXC

Syntax: LEVEL=*n*

where *n* is a number in the range 1–4.

### **LIMIT=**

Purpose: Specifies the size limit of a data set in megabytes.

Allowed in: Rule SET parameter for function SPACLIMI

Syntax: LIMIT=*nnnnnnnnnnK,M,G,T*

where *nnnnnnnnnn* is 1–10 digits and K,M,G,T specifies whether the number is expressed in kilobytes, megabytes, gigabytes, or terabytes. The maximum allowable specification for LIMIT is 2147483647K

### **LLQ=**

Purpose: Specifies or contains the low-level qualifier of a data set name. MAINVIEW SRM name masking can be used for filter list entries.

Allowed in: INC and EXC and rule SET parameter for function DSNCHECK

Syntax: LLQ=*xxxxxxx*

where *xxxxxxx* is a valid data set name qualifier 1–8 characters long.

### **LRECL=**

Purpose: Contains the logical record length of a data set.

Allowed in: INC and EXC

Syntax: LRECL=*<>nnnnn*

where *nnnnn* is a number in the range 1–32,760. The comparison operator symbol can be equals (=), greater than (>), or less than (<).

**MAXQLF=**

Purpose: Specifies the maximum number of qualifiers a data set name can have.

Allowed in: Rule SET parameter for function DSNCHECK

Syntax: MAXQLF=*nn*

where *nn* is a number in the range 1–99.

**MAXSIZE=**

Purpose: Contains the maximum size of a data set in bytes, based on the maximum possible extent count. For a non-VSAM data set, this is the primary extent size plus the size of 15 secondary extents; for VSAM data sets, this is the primary extent size plus the size of 122 secondary extents.

Allowed in: INC and EXC

Syntax: MAXSIZE=<>*nnnnnnnnnnK,M,G,T*

where *nnnnnnnnnn* is 1–10 digits and K,M,G,T specifies whether the number is expressed in kilobytes, megabytes, gigabytes, or terabytes. The comparison operator symbol can be equals (=), greater than (>), or less than (<). The maximum allowable specification for MAXSIZE is 2147483647K.

**MGMTCLAS=**

Purpose: Specifies or contains the name of a DFSMS management class. MAINVIEW SRM name masking can be used for filter list entries. Rule list entries must specify a valid management class name.

Allowed in: INC and EXC and rule SET parameter for function SMSACSMC

Syntax: MGMTCLAS=*xxxxxxxx*

where *xxxxxxxx* is a valid management class name 1–8 characters long.

**MIGCMD=**

Purpose:	Specifies whether DFHSM migration initiated by command (instead of automatically) is to be handled by HSMMIGRT. If MIGCMD=Y is set, the MAINVIEW SRM function HSMMIGRT is activated for migration initiated automatically and on command. If MIGCMD=N, HSMMIGRT is activated only for automatic migration processing.
Allowed in:	Rule SET parameter for function HSMMIGRT
Syntax:	MIGCMD= Y/N
Default:	MIGCMD=Y

**MIGDAYS=**

Purpose:	Specifies the interval in days that a data set must be unreferenced before being eligible for migration. This count is added to the data set's date of last reference; if the resultant date is less than or equal to the current date, the data set is marked eligible for migration. Also see the CAL parameter.  <b>Note:</b> The MIGDAYS parameter is no longer valid for HSMMIGRT. It has been replaced by CALAGE and REFAGE. MIGDAYS remains available for compatibility only; it will cause no action.
Allowed in:	Rule SET parameter for function HSMMIGRT
Syntax:	MIGDAYS= <i>nnnn</i>  where <i>nnnn</i> is a number in the range 1–9999.

**MIGRATE=**

Purpose:	Specifies that DFHSM migration should be allowed or disallowed. When DFHSM tries to migrate data sets, the MAINVIEW SRM function HSMMIGRT gets control. Migration can be allowed or disallowed from HSMMIGRT by setting MIGRATE to Y or N respectively.
Allowed in:	Rule SET parameter for function HSMMIGRT
Syntax:	MIGRATE= Y/N
Default:	MIGRATE=Y



**MINQLF=**

Purpose: Specifies the minimum number of qualifiers a data set name can have.

Allowed in: Rule SET parameter for function DSNCHECK

Syntax: MINQLF=*nn*

where *nn* is a number in the range 1–99.

**ML2=**

Purpose: Specifies that DFHSM migration from ML0 direct to ML2 (skipping ML1) is allowed. (DFHSM release 2.4.0 or higher is required.) Also see the BACKUP parameter.

Allowed in: Rule SET parameter for function HSMMIGRT

Syntax: ML2= Y/N

Default: ML2=N

**MNTYPE=**

Purpose: Contains and specifies the mount type for additional volumes in secondary space allocation. MNTYPE=ALL considers all volumes in the pool, regardless of the volume mount type. MNTYPE=CURRENT searches only for volumes with the same mount type as the current volume. Note that, regardless of the MNTYPE specification, all volumes in an alternate pool are considered eligible.

Also, for compatibility with STOP-X37 comparison, triplets can be specified with the first operand in the triplet specifying a partial mount type, the second operand specifying the offset in the mount type for the comparison to start, and the third operand, the comparison operator. For example, MNTYPE=((PV,1,EQ),(ST,1,EQ)) would allow volumes that are mounted PUBLIC or STORAGE. Valid operators are

EQ	=	NE	¬=
GT	>	LT	<
LE	<=	GE	>=

Allowed in: INC and EXC and rule SET parameter for function SPACVOLA

**Note:** The mount attribute is not available if no volume has been selected.

Syntax: MNTYPE=xxxxxxx

where *xxxxxxx* is the mount status value from the following list:

ALL	All volumes in pool, regardless of mount type
CURRENT	Current volume
PRIVATE	Private volume
PUBLIC	Public volume
STORAGE	Storage volume

Default:

ALL

**Note:** When used as a selection parameter (INC and EXC), MNTYPE will never contain ALL.

or

MNTYPE=((*mmmmmm*,*n*,*op*),...)

where *mmmmmm* is the comparison character string, *n* is the comparison offset, and *op* is the comparison operator.

## MODE=

Purpose:

MODE is the FLST SET statement parameter that specifies the status of the function for the resources that are selected by following INC and EXC statements. When SET MODE=INACT is specified, any selected resources are bypassed for processing by the function. When SET MODE=SIM is specified, selected resources are processed in simulation mode, in which the action is not applied but a message is issued (depending on the MSG parameter) to indicate the action that would be applied if SET MODE=ACT were specified.

**Note:** If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

The MODE parameter is the most significant filter list SET statement parameter. The MODE parameter defines the processing mode for the selected resources. The MODE parameter can be set to one of the following values:

ACTive	(the function acts on the selected resources)
INACTive	(the function does nothing for the selected resources)
SIMulate	(the function reports activity as if it were active, but it does not actually take any action for the selected resources)

Different sets of selected resources can have different processing modes for a function.

**Tip:** Simulate mode (SIM) is used only for the HSM collector, Allocation, and DMS2HSM.

Syntax: MODE=[ACT | INACT | SIM]

Required: Yes

Default: None

**MSG=**

**Purpose:** MSG is the FLST SET statement parameter that specifies the message generation option for resources that are selected by the following parameters. Informational and error messages can be produced, or all messages can be suppressed. Note that the MSG option on the function definition in the SMFUNC $_{xx}$  member is overridden by this option for specific selected resources.

**Note:** The MSG parameter is used only for the HSM collector, Allocation, and DMS2HSM.

Syntax: MSG=[I | W | E | S | N]

I	Informational and error messages
W	Warning messages
E	Error messages only
S	Severe messages
N	No messages

Required: No

Default: The MSG parameter on the function definition in SMFUNC $_{xx}$ .

**NEWAPPL=**

**Purpose:** Specifies the value of an account code that is used to override the default account code. NEWAPPL replaced NEWACCT, which is also still accepted.

Allowed in: Rule SET parameter for function SGCONTRL

Syntax: NEWAPPL=xxxxxxx

where xxxxxx is an alphanumeric code 1–50 characters long.

**NOCATLG2=**

**Purpose:** Specifies the action to be taken when a NOT CATLGD2 condition occurs. This condition occurs when a data set has a disposition of (NEW,CATLG) and the same name already exists in the catalog. The NOCATLG2 function can cancel the job, or rename or delete or uncatalog the old data set. Also see the parameter PURGE.

**Allowed in:** Rule SET parameter for function NOCATLG2

**Syntax:** NOCATLG2=xxxxxxx

where xxxxxx is a value from the following list:

FAIL	Causes the job to fail.
RENAME	Renames the old data set.
DELETE	Deletes the old data set.
UNCATLG	Uncatalogs the old data set.
CANCEL	Cancels the job. If NOCATWHEN=ALLOC, the job is canceled before the current step executes. If NOCATWHEN=TERM, the job is canceled after the current step ends, that is, all following steps are flushed.
NO	The jobstep completes with a normal return code. Subsequent jobsteps might abend or process invalid data. For SMS-managed data sets, the job fails immediately with a JCL error.
FLUSH	The data set receives NOT CATLGD2 message; the remainder of the jobsteps are flushed.
OPER	Issues a message to the system console by way of WTOR, allowing the operator to reply with the desired option.

**NOCATWHEN=**

**Purpose:** Specifies when NOCATLG2 processing will occur for a non-SMS data set.

**Allowed in:** The SET parameter in function NOCATLG2

**Syntax:** NOCATWHEN=TERM/ALLOC

TERM NOCATLG2 processing will occur during step termination.

ALLOC NOCATLG2 will occur during OS/390 data set allocation processing.

**Note:** When NOCATWHEN=ALLOC and no volser is specified, you will not be able to filter on parameter VOL=. To be able to filter on VOL=, you must specify NOTCATWHEN=TERM.

**NOCHECK=**

**Purpose:** Specifies the checks bypassed in space recovery validation.

**Allowed in:** Rule SET parameter for functions SPACSECA, SPACSECB, SPACSECR, SPACSWIR, and SPACVOLA. All of the options are valid for SPACVOLA. CONTIG is the only option valid for the other functions.

**Syntax:** NOCHECK=(*xxxxxx*,...)

where *xxxxxx* is one or more values from the following list:

CONTIG	A data set is allocated with contiguous space required.
DC	A data set resides on a cached device. Under normal conditions, the volume switch occurs only to packs that have the same device characteristics.
DSNAME	A data set is allocated to another DD statement within the same jobstep.
DISP	A permanent data set is being accessed without the use of a catalog.
ENQ	A permanent data set is allocated to a DD statement within another job.
EXCP	A data set is being processed with the EXCP access method (or otherwise processing at the hardware level).
NOTE	A data set is being processed with the NOTE macro.
POINT	A data set is being processed with the POINT macro.

**NQUAL=**

**Purpose:** Contains the total number of qualifiers in a data set name.

**Allowed in:** INC and EXC

**Syntax:** NQUAL=<>*nn*

where *nn* is a number in the range 1–20. The comparison operator symbol can be equals (=), greater than (>), or less than (<).

**NUNIT=**

**Purpose:** Specifies the number of units requested. This is the larger of units coded UNIT=(SYSALLDA,*n*) or volumes requested VOL=SER=.

**Allowed in:** INC and EXC

**Syntax:** NUNIT=*nn*

where *nn* is a number in the range 1–59.

**NVOL=**

Purpose: Specifies the number of volumes that can be allocated to or requested for a non-VSAM data set or VSAM data component

Allowed in: INC and EXC and rule SET parameter for DASDPOOL

Syntax: NVOL=*nn*

where *nn* is a number in the range 1–59 for non-VSAM data sets and 1–20 for VSAM data components

**NVOLINDX=**

Purpose: Specifies the number of volumes that can be allocated to a VSAM index component. Note that CANDIDATE overrides NVOLINDX.

Allowed in: Rule SET parameter for DASDPOOL

Syntax: NVOLINDX=*nn*

where *nn* is a number in the range 1–20.

**NVOLMAX=**

Purpose: Specifies whether the number of volumes that can be assigned to a data set is limited to the number of volumes in the pool in which the data set resides.

Allowed in: Rule SET parameter for DASDPOOL

Syntax: NVOLMAX= *Y/N*

**OLDACCT=**

Purpose: Specifies the value of a default account code.

Allowed in: INC and EXC

Syntax: OLDACCT=*xxxxxxx*

where *xxxxxxx* is an alphanumeric code 1–50 characters long.

**OLDDSN=**

Purpose: Contains the name of the old data set (on a rename operation).

Allowed in: INC and EXC

Syntax: OLDDSN=xxxxxxx.

where xxxxxx is the old data set name up to 44 characters long.

**OLDHLQ=**

Purpose: Contains the old data set high-level qualifier (on a rename operation).

Allowed in: INC and EXC

Syntax: OLDHLQ=xxxxxxx

where xxxxxx is the old data set high-level qualifier.

**OPER=**

Purpose: Allows the operator to provide a volume when the system cannot find space for a volume switch during SPACVOLA. This option is not available for SMS-managed data sets.

Allowed in: Rule SET parameter for function SPACVOLA

Syntax: OPER= Y/N

**ORIGUNIT=**

Purpose: Contains the original unit name (generic or esoteric) specified in the JCL. For VSAM allocations, the value will always be SYSALLDA.

Allowed in: INC and EXC. Cannot be set for SPACPRIM, SPACSECA, SPACSECB, SPACSECR, SPACSWIR, or SPACVOLA.

Syntax: ORIGUNIT=xxxxxxx

where xxxxxx is a unit name.

### **ORIGVOL=**

Purpose: Contains the original VOLSER specified in the JCL or in the IDCAMS control cards.

Allowed in: INC and EXC

Syntax: ORIGVOL=xxxxxx  
where xxxxxx is a volsr.

### **OWNER=**

Purpose: Contains and specifies an owner to be assigned during cluster definition.

Allowed in: INC and EXC and rule SET parameter for function VSAMCNTL. Cannot be set for NOCATLG2, SPACSWIR, or SPACPRIM functions.

Syntax: OWNER=xxxxxxxx  
where xxxxxxxx is a string up to 40 characters long.

### **PCTI=**

Purpose: Specifies the percentage value by which a secondary allocation is increased.

Allowed in: Rule SET parameter for function SPACSECI and SPACVOLA

Syntax: PCTI=nnnnn  
where nnnnn is a number between 0 and 10000

### **PGM=**

Purpose: Contains the name of the currently executing program. MAINVIEW SRM name masking can be used.

Allowed in: INC and EXC

Syntax: PGM=xxxxxxx  
where xxxxxxx is a valid program name 1–8 characters long.

### **PGMRNAME=**



**Purpose:** Contains the value of the programmer name field of the job card of the currently executing job. MAINVIEW SRM name masking can be used.

**Note:** This parameter is not available for functions DSNCHECK, SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, or SMSACSTE.

**Allowed in:** INC and EXC

**Syntax:** PGMRNAME=xxxxxxx

where xxxxxxx is a valid programmer name 1–20 characters long.

### **POOL=**

**Purpose:** Specifies or contains the name of a pool. MAINVIEW SRM name masking can be used for filter list entries. Up to 15 values of this parameter can be specified within parentheses when used as an action parameter on a SET statement.

**Allowed in:** INC and EXC, and rule SET parameter for functions DASDPOOL, HSMRECAL, TAPEPOOL

**Syntax:** POOL=xxxxxxx or POOL=(xxxxxxx,xxxxxxx,...)

where xxxxxxx is a valid pool name 1–8 characters long. If the pool name is not defined, refresh or startup will fail. Up to 15 pool names can be specified in parentheses.

### **PQTY=**

**Purpose:** Specifies the size in kilobytes for the primary space allocation.

**Allowed in:** Rule SET parameter for function SPACSQTY

**Syntax:** PQTY=nnnnnK,M,G,T

where nnnnn is a number in the range 1–99999K.

### **PRISPACE=**

**Purpose:** Contains the requested primary space in the units specified in the space request.

**Allowed in:** INC and EXC

**Syntax:** PRISPACE=*nnnnnn*  
where *nnnnnn* is a number in the range 0–999999.

### **PROCSTEP=**

**Purpose:** Contains the step name of the currently executing procedure. MAINVIEW SRM name masking can be used.

**Note:** This parameter is not available for functions DSNCHECK, SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, or SMSACSTE.

**Allowed in:** INC and EXC

**Syntax:** PROCSTEP=*xxxxxxxx*  
where *xxxxxxxx* is a character string 1–8 bytes long.

### **PURGE=**

**Purpose:** Specifies whether the old data set should be purged when a NOT CATLG2 error occurs and NOCATLG2=DELETE and the old data set has an unexpired expiration date. Also see the parameter NOCATLG2.

**Allowed in:** Rule SET parameter for function NOCATLG2

**Syntax:** PURGE= Y/N

### **PWDDEL=**

**Purpose:** Specifies that any passwords specified for a VSAM definition be deleted.

**Allowed in:** Rule SET parameter for function VSAMCNTL

The PWDDEL=Y is specified if all password specifications (CONTROLPW, MASTERPW, READPW, UPDATEPW) are removed from the cluster definition; if PWDDEL=N is specified, any existing password specification is retained.

**Syntax:** PWDDEL= Y/N

**QUALn=**

Purpose:                   Synonym. See DSNn.

**QUALL=**

Purpose:                   Synonym. See LLQ.

**RACF=**

Purpose:                   Contains the name of the RACF or CA-Top Secret group. MAINVIEW SRM name masking can be used. (This parameter will not have a value if your security system is CA-ACF2.)

Allowed in:            INC and EXC

Syntax:                RACF=xxxxxxxx

**RACFGRP=**

Purpose:                   Tests the value of either the RACF group coded on the JOB card or the default RACF group.

Allowed in:            INC and EXC

Syntax:                RACFGRP=xxxxxxxx  
  
                          where xxxxxxxx is a user ID 1–8 characters long.

**RACFUID=**

Purpose:                   Specifies the value of the RACF user ID on a JOB card.

Allowed in:            INC and EXC

Syntax:                RACFUID=xxxxxxxx  
  
                          where xxxxxxxx is a user ID 1–8 characters long.

**RAIDDEVTYPE=**

**Purpose:** Allows the user to require a specific RAID device type for an allocation. If a particular hardware is chosen, the candidate list of volumes will be limited to devices that have the requested attribute flag enabled.

**Allowed in:** Rule SET parameter for functions DASDPOOL and SMSSELECT

**Syntax:** RAIDDEVTYPE=xxxxxxxx

where xxxxxxxx is a value from the following list:

EMC	The device must be an EMC RAID device.
RDFEMC	The device must be EMC and have the RDF flag enabled.
MIRROREMC	The device must be EMC and have the MIRROR flag enabled.
PARITYEMC	The device must be EMC and have the PARITY flag enabled.

**RECFM=**

**Purpose:** Contains the record format of a data set.

**Allowed in:** INC and EXC

**Syntax:** RECFM=xxx

where the first *x* is a value from the following list:

F	Fixed
V	Variable
U	Undefined

and the second and third letters are

B	Blocked
S	Spanned
T	Track overflow
M	Machine control character
A	ANSI control character

**RECORD=**

Purpose: Contains the record organization of a VSAM data set.

Allowed in: INC and EXC

Syntax: RECORD=*xx*

where *xx* is a value from the following list:

RR	Relative record
ES	Entry sequenced
KS	Key sequenced
LS	Linear

**REFAGE=**

Purpose: Contains the unadjusted, unreferenced day count set by the HSMMIGRT.

Allowed in: INC and EXC

Syntax: REFAGE=*nnnn*

where *nnnn* is a number in the range 0—9999

**REFVOL=**

Purpose: Contains the volume serial number of the referenced DASD volume. This is the DASD volume containing the data set referenced by a VOL=REF parameter in the JCL DD statement.

**Note:** If the VOL=REF refers to an uncataloged data set name, REFVOL will contain the string NULVRF. Referring to an uncataloged data set in a VOL=REF statement will normally cause a JCL error; however, this JCL error can be suppressed by the SUPVOLRF function.

Allowed in: INC and EXC parameter for the functions: SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, and DASDPOOL

Syntax: REFVOL=*xxxxxx*

### **REJECT=**

Purpose: Specifies whether the OS/390 service request is to be rejected or accepted.

Allowed in: Rule SET parameter for functions DASDPOOL, SMSACSDC, and TAPEPOOL

Syntax: REJECT= Y/N

### **RELEASE=**

Purpose: Contains the space release flag.

Allowed in: INC and EXC

Syntax: RELEASE= Y/N

### **REORG=**

Purpose: Specifies whether SPACVOLA will start a started task to reorganize a data set that has just been made multivolume by SPACVOLA.

Allowed in: Rule SET parameter for function SPACVOLA

Syntax: REORG=Y/N

### **REORG\_NSMS=**

Purpose: Specifies the SMRORGxx suffix that contains the DFDSS control cards to be used by the DFDSS reorganize started task to reorganize a non-SMS multivolume data set. Also specifies the MAINVIEW SRM pool name of the target pool to which the multivolume data set is to be reorganized.

Allowed in: Rule SET parameter for function SPACVOLA

Syntax: REORG\_NSMS=(*xx,poolname*)

where *xx* is the SMRORGxx suffix and *poolname* is the MAINVIEW SRM target pool for the reorganize.

**REORG\_PROC=**

**Purpose:** Specifies the name of the procedure library member to be used as the started task JCL for the DFDSS reorganize job that is started by SPACVOLA processing automatically if REORG=Y is specified in the SPACVOLA RLST SET statement.

**Allowed in:** Rule SET parameter for function SPACVOLA

**Syntax:** REORG\_PROC=(*procname*)

where *procname* is the procedure library member to use as the started task JCL for the reorganize job.

**REORG\_SMS=**

**Purpose:** Specifies the SMRORGxx suffix that contains the DFDSS control cards to be used by the DFDSS reorganize started task to reorganize an SMS multivolume data set. Also specifies the DFSMS Storage Class to be used as the target Storage Class to which the multivolume data set is to be reorganized.

**Allowed in:** Rule SET parameter for function SPACVOLA

**Syntax:** REORG\_SMS=(*xx,storclas*)

where *xx* is the SMRORGxx suffix and *storclas* is the DFSMS target Storage Class for the reorganize.

**REPL=**

**Purpose:** Contains and specifies the removal of the REPLICATE parameter during cluster definition.

**Allowed in:** INC and EXC and rule SET parameter for function VSAMCNTL

**Syntax:** REPL=Y/N

If Y is specified, the REPLICATE parameter is forced in the DEFINE CLUSTER definition; if N is specified, the REPLICATE parameter is removed from the DEFINE CLUSTER definition.

**Note:** When used as a selection parameter, this parameter is only valid for VSAM data sets.

**REPLACE=**

Purpose:	Specifies that the OS/390 value is to be replaced by the MAINVIEW SRM calculated value.
Allowed in:	Rule SET parameter for functions SETEXPDT, SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, and SPACSQTY
Syntax:	REPLACE= Y/N

**RETPD=**

Purpose:	Contains and specifies the retention period in days for a new data set.
Allowed in:	INC and EXC and rule SET parameter for function SETEXPDT
Syntax:	RETPD=nnnn  where <i>nnnn</i> is a number in the range 0–9999.

**REUSE=**

Purpose:	Contains and specifies the removal of the REUSE parameter during cluster definition.
Allowed in:	INC and EXC and rule SET parameter for function VSAMCNTL
Syntax:	REUSE= Y/N  If Y is specified, the REUSE parameter is forced in the DEFINE CLUSTER definition; if N is specified, the REUSE parameter is removed from the DEFINE CLUSTER definition.  <b>Note:</b> When used as a selection parameter, this parameter is only valid for VSAM data sets.

**RLSE=**

Purpose:	Specifies that the SPACRLSE function is to release space for newly allocated data sets.
Allowed in:	Rule SET parameter for function SPACRLSE
Syntax:	RLSE=ALL/SEC/NO  ALL    Releases space for all data sets SEC    Releases space for data sets with a secondary allocation NO     Turns the release flag off



**ROUND=**

Purpose:	Contains and specifies that the ROUND subparameter of the SPACE parameter is in the JCL.
Allowed in:	INC and EXC and rule SET parameter for functions SPACCONV and SPACSQTY
Syntax:	ROUND= Y/N

**SECSPACE=**

Purpose:	Contains the requested secondary space in the units specified in the space request.
Allowed in:	INC and EXC
Syntax:	SECSPACE= <i>nnnnnn</i> K,M,G,T  where <i>nnnnnn</i> is a number in the range of 0–999999.

**SCAN=**

Purpose:	Specifies not to budget space for any level associated with a data set; this parameter is unique to the application collector
Allowed in:	Rule SET parameter
Syntax:	SCAN=EXIT

**SEP=**

Purpose:	Specifies whether the data and index components of a VSAM key-sequenced data set are allocated to separate volumes in a pool.  CANDIDATE=Y must also be specified for VSAM component separation.
Allowed in:	Rule SET parameter for function DASDPOOL
Syntax:	SEP= Y/N/ASIS

**SGC\_FUNC=**

Purpose: Specifies the value of the application collector function currently being processed.

Allowed in: INC and EXC

Syntax: SGC\_FUNC=xxxxxxx

where xxx is a valid value from the following list:

ALLOCATE  
EXTENDCV (extend on current volume)  
EXTENDNV (extend on new volume)  
EXTENDVS (extend on VSAM)  
RELEASE  
RENAME  
SCRATCH  
BUDGET (TSO command being executed)  
BUDDSN (TSO command being executed)  
SGCMAINT (program being executed)  
SGCRSYNC (program being executed)  
SGCHSMR (batch HSM report program is being executed)  
SVOSISPF (SGC programs are being invoked from the SVOS ISPF interface)

Default: None

**SGDA\_ALNV=**

Purpose: Specifies the total space allocated to non-VSAM data sets in the account. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

Allowed in: INC and EXC

Syntax: SGDA\_ALNV=nnnnnnnnnn

where nnnnnnnnnn is a number of 64K units between 1 and 2147483647

Default: None

**SGDA\_ALV=**

**Purpose:** Specifies the total space allocated to VSAM data sets in the account. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

**Allowed in:** INC and EXC

**Syntax:** SGDA\_ALV=*nnnnnnnnnn*

where *nnnnnnnnnn* is a number of 64K units between 1 and 2147483647

**Default:** None

**SGDA\_AVAIL=**

**Purpose:** Specifies the total space available in the account. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

**Allowed in:** INC and EXC

**Syntax:** SGDA\_AVAIL=*nnnnnnnnnn*

where *nnnnnnnnnn* is a number of 64K units between 1 and 2147483647

**Default:** None

**SGDA\_GRP=**

**Purpose:** Specifies the application group name; also known as account name.

**Allowed in:** INC and EXC

**Syntax:** SGDA\_GRP=*xxxxxxxx...*

where *xxxxxxxx* is a 1 to 50 character group name.

**Default:** None

**SGDA\_IDLE=**

Purpose: Specifies the total allocated space that is unused in the account. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

Allowed in: INC and EXC

Syntax: `SGDA_IDLE=nnnnnnnnnnnn`  
where *nnnnnnnnnnnn* is a number of 64K units between 1 and 2147483647

Default: None

**SGDA\_NVDS=**

Purpose: Specifies the number non-VSAM data sets in the account. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

Allowed in: INC and EXC

Syntax: `SGDA_NVDS=nnnnnn`  
where *nnnnnn* a number between 1 and 65535

Default: None

**SGDA\_VSD=**

Purpose: Specifies total number of VSAM data sets in the account. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

Allowed in: INC and EXC

Syntax: `SGDA_VSD=nnnnnn`  
where *nnnnnn* is a number between 1 and 65535

Default: None

**SGDP\_ALNV=**

**Purpose:** Specifies the space allocated for non-VSAM data sets in the pool. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

**Allowed in:** INC and EXC

**Syntax:** `SGDP_ALNV=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number between 1 and 2147483647

**Default:** None

**SGDP\_ALV=**

**Purpose:** Specifies the total space allocated to VSAM data sets in the pool. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

**Allowed in:** INC and EXC

**Syntax:** `SGDP_ALV=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number between 1 and 2147483647

**Default:** None

**SGDP\_AVAIL=**

**Purpose:** Specifies the total space available in the pool. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

**Allowed in:** INC and EXC

**Syntax:** `SGDP_AVAIL=nnnnnnnnnn`  
where *nnnnnnnnnn* is a number between 1 and 2147483647

**Default:** None

**SGDP\_DS#DA=**

Purpose: Specifies the total number of data sets on volumes with a SMS status of DISABLED/ALL.

Allowed in: INC and EXC

Syntax: SGDP\_DS#DA=*nnnnnnnnn*  
where *nnnnnnnnn* is the number of data sets from 0 to 16777215

Default: None

**SGDP\_DS#DN=**

Purpose: Specifies the total number of data sets on volumes with a SMS status of DISABLED/NEW.

Allowed in: INC and EXC

Syntax: SGDP\_DS#DN=*nnnnnnnnn*  
where *nnnnnnnnn* is the number of data sets from 0 to 16777215

Default: None

**SGDP\_DS#EN=**

Purpose: Specifies the total number of data sets on volumes with an SMS status of ENABLED.

Allowed in: INC and EXC

Syntax: SGDP\_DS#EN=*nnnnnnnnn*  
where *nnnnnnnnn* is the number of data sets from 0 to 16777215

Default: None

**SGDP\_DS#NM=**

Purpose: Specifies the total number of data sets on volumes with an SMS status of NOT SMS MANAGED.

Allowed in: INC and EXC

Syntax: SGDP\_DS#NM=*nnnnnnnnn*

where *nnnnnnnnn* is the number of data sets from 0 to 16777215

Default: None

**SGDP\_DS#NS=**

Purpose: Specifies the total number of data sets on volumes with an SMS status of NO STATUS GIVEN.

Allowed in: INC and EXC

Syntax: SGDP\_DS#NS=*nnnnnnnnn*

where *nnnnnnnnn* is the number of data sets from 0 to 16777215

Default: None

**SGDP\_DS#QA=**

Purpose: Specifies the total number of data sets on volumes with an SMS status of QUIESCED/ALL.

Allowed in: INC and EXC

Syntax: SGDP\_DS#QA=*nnnnnnnnn*

where *nnnnnnnnn* is the number of data sets from 0 to 16777215

Default: None

### **SGDP\_DS#QN=**

Purpose: Specifies the total number of data sets on volumes with an SMS status of QUIESCED/NEW.

Allowed in: INC and EXC

Syntax: **SGDP\_DS#QN=nnnnnnnnn**  
 where *nnnnnnnnn* is the number of data sets from 0 to 16777215

Default: None

### **SGDP\_FRAGI=**

Purpose: Specifies the fragmentation index.

Allowed in: INC and EXC

Syntax: **SGDP\_FRAGI=nnnn**  
 where *nnnn* is a value between 0 and 1000

Default: None

### **SGDP\_IDLE=**

Purpose: Specifies the space allocated and unused in the pool. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

Allowed in: INC and EXC

Syntax: **SGDP\_IDLE=nnnnnnnnnnnn**  
 where *nnnnnnnnnnnn* is a number between 1 and 2147483647

Default: None

### **SGDP\_NCLPER=**

Purpose: Specifies the net capacity load percentage in tenths of a percent.

Allowed in: INC and EXC

Syntax: **SGDP\_NCLPER=nnnn**  
 where *nnnn* is a number between 0 and 1000



**SGDP\_NNV=**

Purpose: Specifies the number of non-VSAM data sets in the pool.

Allowed in: INC and EXC

Syntax: **SGDP\_NNV=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

**SGDP\_NV=**

Purpose: Specifies the number of VSAM data sets in the pool.

Allowed in: INC and EXC

Syntax: **SGDP\_NV=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

**SGDP\_NVOL=**

Purpose: Specifies the number of volumes in the pool.

Allowed in: INC and EXC

Syntax: **SGDP\_NVOL=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

**SGDP\_OFFL=**

Purpose: Specifies the total offline volumes offline.

Allowed in: INC and EXC

Syntax: **SGDP\_OFFL=nnnnn**

where *nnnnn* is a numeric value between 0 and 65535

Default: None

**SGDP\_PERFUL=**

Purpose: Specifies the Percentage Full or Percentage Allocated for all volumes in the pool.

Allowed in: INC and EXC

Syntax: **SGDP\_PERFUL=***nnn*  
where *nnn* is a number between 0 and 100

**SGDP\_POOL=**

Purpose: Specifies the pool name for reporting.

Allowed in: INC and EXC

Syntax: **SGDP\_POOL=***xxxxxxxx*  
where *xxxxxxxx* is a 1 to 8 character pool name.

Default: None

**SGDP\_RSVD=**

Purpose: Specifies the total reserved space in the pool. This number is in 64KB units, where a value of one is equal to 65,536 bytes.

Allowed in: INC and EXC

Syntax: **SGDP\_RSVD=***nnnnnnnnnn*  
where *nnnnnnnnnn* is a number between 1 and 2147483647

Default: None

**SGDP\_RVAARC=**

**Purpose:** Specifies the array capacity of the device for RVA pools in tenths of megabytes in scale. For example, a value of one in the field indicates 0.1 of a megabyte, a value of 10 indicates 1.0 megabytes, and so on.

**Allowed in:** INC and EXC

**Syntax:** SGDP\_RVAARC=*nnnnnnnnnn*

where *nnnnnnnnnn* is a number between 1 and 2147483647

**Default:** None

**SGDP\_RVAFNC=**

**Purpose:** Specifies the amount of space not collected by free space collection activity during the interval for RVA pools in tenths of megabytes in scale. For example, a value of one in the field indicates 0.1 of a megabyte, a value of 10 indicates 1.0 megabytes, and so on.

**Allowed in:** INC and EXC

**Syntax:** SGDP\_RVAFNC=*nnnnnnnnnn*

where *nnnnnnnnnn* is a number between 1 and 2147483647

**Default:** None

**SGDP\_RVAFSC=**

**Purpose:** Specifies the amount of space collected by free space collection activity during the interval for RVA pools in tenths of megabytes in scale. For example, a value of one in the field indicates 0.1 of a megabyte, a value of 10 indicates 1.0 megabytes, and so on.

**Allowed in:** INC and EXC

**Syntax:** SGDP\_RVAFSC=*nnnnnnnnnn*

where *nnnnnnnnnn* is a number between 1 and 2147483647

**Default:** None

**SGDP\_RVAIND=**

Purpose:	Specifies whether the pool is for an RVA device (see SGDP_PTYPE of V.)
Allowed in:	INC and EXC
Syntax:	SGDP_RVAIND=Y/N
Default:	None

**SGDP\_RVANCL=**

Purpose:	Specifies the net capacity load of the RVA device in tenths of megabytes in scale. For example, a value of one in the field indicates 0.1 of a megabyte, a value of 10 indicates 1.0 megabytes, and so on.
Allowed in:	INC and EXC
Syntax:	SGDP_RVANCL= <i>nnnnnnnnnn</i>  where <i>nnnnnnnnnn</i> is a number between 1 and 2147483647
Default:	None

**SGDP\_SPACDA=**

Purpose:	Specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of DISABLED/ALL.
Allowed in:	INC and EXC
Syntax:	SGDP_SPACDA= <i>nnnnnnnnnn</i>  where <i>nnnnnnnnnn</i> is a numeric value between 0 and 2147483647
Default:	None

**SGDP\_SPACDN=**

Purpose:	Specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of DISABLED/NEW.
Allowed in:	INC and EXC
Syntax:	SGDP_SPACDN= <i>nnnnnnnnnn</i>  where <i>nnnnnnnnnn</i> is a numeric value between 0 and 2147483647

Default: None

**SGDP\_SPACEN=**

Purpose: Specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of ENABLED.

Allowed in: INC and EXC

Syntax: SGDP\_SPACEN=*nnnnnnnnnn*

where *nnnnnnnnnn* is a numeric value between 0 and 2147483647

Default: None

**SGDP\_SPACNM=**

Purpose: Specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of NOT SMS MANAGED.

Allowed in: INC and EXC

Syntax: SGDP\_SPACNM=*nnnnnnnnnn*

where *nnnnnnnnnn* is a numeric value between 0 and 2147483647

Default: None

**SGDP\_SPACNS=**

Purpose: Specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of NO STATUS GIVEN.

Allowed in: INC and EXC

Syntax: SGDP\_SPACNS=*nnnnnnnnnn*

where *nnnnnnnnnn* is a numeric value between 0 and 2147483647

Default: None

**SGDP\_SPACQA=**

Purpose: Specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of QUIESCED/ALL.

Allowed in: INC and EXC

Syntax:	SGDP_SPACQA= <i>nnnnnnnnnn</i>  where <i>nnnnnnnnnn</i> is a numeric value between 0 and 2147483647
Default:	None
<b>SGDP_SPACQN=</b>	
Purpose:	Specifies the total space allocated (in 64 megabytes) on volumes with an SMS status of QUIESCED/NEW.
Allowed in:	INC and EXC
Syntax:	SGDP_SPACQN= <i>nnnnnnnnnn</i>  where <i>nnnnnnnnnn</i> is a numeric value between 0 and 2147483647
Default:	None
<b>SGDP_TYPE=</b>	
Purpose:	Specifies the type of pool.
Allowed in:	INC and EXC
Syntax:	SGDP_TYPE= <i>x</i>  where <i>x</i> is one of the following: M - OS/390 esoteric name P - MAINVIEW SRM pool R - RAID pseudo pool S - SMS pool U - User pool V - RVA pseudo pool
Default:	None
<b>SGDP_VOL#DA=</b>	
Purpose:	Specifies the total number of volumes with a SMS status of DISABLED/ALL.
Allowed in:	INC and EXC
Syntax:	SGDP_VOL#DA= <i>nnnnn</i>  where <i>nnnnn</i> is a numeric value between 0 and 65535

Default: None

**SGDP\_VOL#DN=**

Purpose: Total volumes with an SMS status of DISABLED/NEW.

Allowed in: INC and EXC

Syntax: SGDP\_VOL#DN=*nnnnn*

where *nnnnn* is a numeric value between 0 and 65535

Default: None

**SGDP\_VOL#EN=**

Purpose: Specifies the total number of volumes with an SMS status of ENABLED.

Syntax: SGDP\_VOL#EN=*nnnnn*

where *nnnnn* is a numeric value between 0 and 65535

Default: None

**SGDP\_VOL#NM=**

Purpose: Specifies the total number of volumes with an SMS status of NOT SMS MANAGED.

Allowed in: INC and EXC

Syntax: SGDP\_VOL#NM=*nnnnn*

where *nnnnn* is a numeric value between 0 and 65535

Default: None

**SGDP\_VOL#NS=**

Purpose: Specifies the total number of volumes with an SMS status of NO STATUS GIVEN.

Allowed in: INC and EXC

Syntax: SGDP\_VOL#NS=*nnnnn*

where *nnnnn* is a numeric value between 0 and 65535

Default: None

**SGDP\_VOL#QA=**

Purpose: Specifies the total number of volumes with an SMS status of QUIESCED/ALL.

Allowed in: INC and EXC

Syntax: **SGDP\_VOL#QA=nnnnn**  
where *nnnnn* is a numeric value between 0 and 65535

Default: None

**SGDP\_VOL#QN=**

Purpose: Specifies the total number of volumes with an SMS status of QUIESCED/NEW.

Allowed in: INC and EXC

Syntax: **SGDP\_VOL#QN=nnnnn**  
where *nnnnn* is a numeric value between 0 and 65535

Default: None

**SGDV\_ALREXT=**

Purpose: Specifies the number of additional tracks in largest free extent on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_ALREXT=nnnnn**  
where *nnnnn* is a number between 1 and 65535

Default: None

**SGDV\_FRAGI=**

Purpose: Specifies the fragmentation index on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_FRAGI=nnnnn**



where *nnnnn* is a number between 1 and 1000

Default: None

**SGDV\_FRCYL=**

Purpose: Specifies the number of free cylinders on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_FRCYL=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

**SGDV\_FREXT=**

Purpose: Specifies the number of free extents on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_FREXT=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

**SGDV\_FRVIR=**

Purpose: Specifies the free VIR count on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_FRVIR=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

**SGDV\_IDTR=**

Purpose: Specifies the total number of idle tracks on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_IDTR=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

#### **SGDV\_LREXT=**

Purpose: Specifies the number of cylinders in largest free extent on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_LREXT=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

#### **SGDV\_LREXTT=**

Purpose: Specifies the size of largest extent in tracks on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_LREXTT=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

#### **SGDV\_NDS=**

Purpose: Specifies the total number of data sets on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_NDS=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default: None

#### **SGDV\_NF0DSC=**

Purpose: Specifies the format 0 (free) DSCB count on the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_NF0DSC=nnnnn**

where *nnnnn* is a number between 1 and 65535

Default:               None

**SGDV\_PERFUL=**

Purpose:                Specifies the Percentage Full or Percentage Allocated for the volume.

Allowed in:           INC and EXC

Syntax:               SGDV\_PERFUL=*nnn*

where *nnn* is a number between 0 and 100

**SGDV\_PHYID=**

Purpose:                Specifies the physical disk ID.

Allowed in:           INC and EXC

Syntax:               SGDV\_PHYID=*xxxxxxxxxx*

where *xxxxxxxxxx* is 1–11 characters that identify the physical disk.

Default:               None

## **SGDV\_PHYIDT=**

Purpose: Specifies the physical disk ID type.

Allowed in: INC and EXC

Syntax: **SGDV\_PHYIDT=x**

where *x* is one of the following 1-character values:

- 0 - Unknown/non-RAID
- A - RAMAC array DASD
- B - RAMAC array subsystem
- C - RAMAC2 array DASD
- D - RAMAC2 array subsystem
- E - RAMAC virtual array
- F - RAMAC3 array DASD
- G - IBM 2105 device
- H - HTC emulated 2105 device
- I - EMC Symmetrix
- J - Emulated 2105 device

Default: None

## **SGDV\_POOL=**

Purpose: Specifies the first pool name in which the volume is defined.

Allowed in: INC and EXC

Syntax: **SGDV\_POOL=xxxxxxx**

where *xxxxxxx* is a 1 to 8 character pool name.

Default: None

## **SGDV\_POOL1=**

Purpose: Specify pool name in which the volume is defined.

Allowed in: INC and EXC

Syntax: **SGDV\_POOL1=xxxxxxx**

where *xxxxxxx* is a 1 to 8 character pool name

Default: None

**SGDV\_PTYP=**

Purpose: Specifies the pool type.

Allowed in: INC and EXC

Syntax: SGDV\_PTYP=*x*

where *x* is one of the following:

M - OS/390 esoteric name

P - MAINVIEW SRM pool

R - RAID pseudo pool

S - SMS pool

U - User pool

V - RVA pseudo pool

Default: None

**SGDV\_RVAIND=**

Purpose: Indicates if the volume exists on a RVA frame. If this value is Y then the other RVA fields can be used.

Allowed in: INC and EXC

Syntax: SGDV\_RVAIND= Y/N

Default: None

**SGDV\_RVAFDV=**

Purpose: Specifies the functional device ID for a volume existing on a RVA frame. This field is blank unless the SGDV\_RVAIND field is Y.

Allowed in: INC and EXC

Syntax: SGDV\_RVAFDV=*xx*

where *xx* is a 1 to 2 character device ID

Default: None

**SGDV\_RVAPCS=**

Purpose: Specifies the physical capacity shared for a volume existing on a RVA device in tenths of megabytes in scale. For example, a value of one in the field indicates 0.1 of a megabyte, a value of 10 indicates 1.0 megabytes, and so on. This field is blank unless the SGDV\_RVAIND field is Y.

Allowed in: INC and EXC

Syntax: `SGDV_RVAPCS=nnnnn`

where *nnnnn* is a number from 1 to 65535.

Default: None

#### **SGDV\_RVAPCU=**

Purpose: Specifies the physical capacity used for a volume existing on a RVA device in tenths of megabytes in scale. For example, a value of one in the field indicates 0.1 of a megabyte, a value of 10 indicates 1.0 megabytes, and so on. This field is blank unless the `SGDV_RVAIND` field is Y.

Allowed in: INC and EXC

Syntax: `SGDV_RVAPCU=nnnnn`

where *nnnnn* is a number from 1 to 65535.

Default: None

#### **SGDV\_RVASSF=**

Purpose: Specifies the RVA subsystem frame name for the RVA frame the volume exists on. This field is blank unless the `SGDV_RVAIND` field is Y.

Allowed in: INC and EXC

Syntax: `SGDV_RVASSF=xxxxxxxx`

where *xxxxxxxx* is a 1 to 8 character subsystem frame name.

Default: None

#### **SGDV\_RVAVOL=**

Purpose: Specifies the descriptive volume name of a volume existing on a RVA frame. This field is blank unless the `SGDV_RVAIND` field is Y.

Allowed in: INC and EXC

Syntax: `SGDV_RVAVOL=xxxxxxxx`

where *xxxxxxxx* is a 1 to 8 character descriptive volume name.

Default: None

**SGDV\_RSRVDT=**

Purpose: Specifies the number of reserved tracks (not included in free space) on the volume.

Allowed in: INC and EXC

Syntax: `SGDV_RSRVDT=nnnnn`  
where *nnnnn* is a number between 1 and 65535

Default: None

**SGDV\_SMALLD=**

Purpose: Specifies the size of the smallest data set on the volume (in kilobytes).

Allowed in: INC and EXC

Syntax: `SGDV_SMALLD=nnnnnnnn`  
where *nnnnnnnn* is a number of kilobytes between 0 and 16777215

Default: None

**SGDV\_SSID=**

Purpose: Specifies the subsystem ID.

Allowed in: INC and EXC

Syntax: `SGDV_SSID=xxxx`  
where *xxxx* is a 4-character subsystem ID value

Default: None

**SGDV\_USEXT=**

Purpose: Specifies the number of used extents on the volume.

Allowed in: INC and EXC

Syntax: `SGDV_USEXT=nnnnn`  
where *nnnnn* is a number between 1 and 65535

Default: None

**SGDV\_VIRPU=**

Purpose: Specifies the VTOC index percentage used.

Allowed in: INC and EXC

Syntax: **SGDV\_VIRPU=***nnn*

where *nnn* is a number between 1 and 100

Default: None

**SGDV\_VIRSZ=**

Purpose: Specifies the VTOC index size in tracks.

Allowed in: INC and EXC

Syntax: **SGDV\_VIRSZ=***nnnnnnnnnn*

where *nnnnnnnnnn* is numeric and must be between 0 and 2147483647

Default: None

**SGDV\_VOL=**

Purpose: Specifies the volume serial number of the volume.

Allowed in: INC and EXC

Syntax: **SGDV\_VOL=***xxxxxxxx*

where *xxxxxxxx* is a 1 to 8 character volume serial number.

Default: None

**SGDV\_VVDSPU=**

Purpose: Specifies the VVDS percentage used.

Allowed in: INC and EXC

Syntax: **SGDV\_VVDSPU=***nnn*

where *nnn* is a percentage 0–100

Default: None



**SGDV\_VVDSSZ=**

Purpose: Specifies the VVDS size in tracks.

Allowed in: INC and EXC

Syntax: **SGDV\_VVDSSZ=nnnnnnnnnn**

where *nnnnnnnnnn* is the number of tracks between 0 and 2147483647

Default: None

**SGP\_@BUSY=**

Purpose: Specifies channel path busy threshold for inclusion or exclusion.

Allowed in: INC and EXC

Syntax: **SGP\_@BUSY{=|<|>}nnn**

Default: None

**SGP\_BESCOLT=**

Purpose: Specifies the collected back-end space in tenths of a MB.

Allowed in: INC and EXC

Syntax: **SGP\_BESCOLT=nnnnnnnnn**

where *nnnnnnnnn* is 1-8 numbers.

Default: None

**SGP\_BESFREE=**

Purpose: Specifies the free back-end space in tenths of a MB.

Syntax: **SGP\_BESFREE=nnnnnnnnn**

where *nnnnnnnnn* is 1-8 numbers.

Default: None

**SGP\_BESTOTL=**

Purpose: Specifies the total back-end space in tenths of a MB.

Syntax: SGP\_BESTOTL=*nnnnnnnn*  
 where *nnnnnnnn* is 1-8 numbers.

Default: None

**SGP\_BESUNCL=**

Purpose: Specifies the uncollected back-end space in tenths of a MB.

Syntax: SGP\_BESUNCL=*nnnnnnnn*  
 where *nnnnnnnn* is 1-8 numbers.

Default: None

**SGP\_CFWHIT@=**

Purpose: Specifies percentage of DFAST reads satisfied by cache threshold.

Syntax: SGP\_CFWHIT@{=|>|<}*nnn*  
 where *nnn* is 0 to 100.

Default: None

**SGP\_CFWPRSC=**

Purpose: Specifies number of CFAST writes reads per-second threshold.

Syntax: SGP\_CFWPRSC{=|>|<}*nnn*

Default: None

**SGP\_CHPID=**

Purpose: Specifies channel paths to be included or excluded.

Syntax: SGP\_CHPID{=|>|<}*chp ID*

Default: None

**SGP\_CNTLUID=**

Purpose: Specifies subsystem IDs of cache controllers to be included or excluded.

Syntax: SGP\_CNTLUID{=|>|<}*subsystem ID*

Default: None

#### **SGP\_CONNTIM=**

Purpose: Specifies the data set connect time threshold in .1 millisecond increments.

Syntax: SGP\_CONNTIM{=|>|<}nnnnn

Default: None

#### **SGP\_CUBSYDL=**

Purpose: Specifies the control unit busy delay threshold in .1 millisecond increments.

Syntax: SGP\_CUBSYDL{=|>|<}nnnnn

Default: None

#### **SGP\_DFWHITE@=**

Purpose: Specifies percentage of DFAST writes satisfied by cache threshold.

Syntax: SGP\_DFWHITE@ {=|>|<}nnn

where nnn is 0 to 100

Default: None

#### **SGP\_DFWPRSC=**

Purpose: Specifies number of DFAST writes per-second threshold.

Syntax: SGP\_DFWPRSC{=|>|<}nnn

Default: None

#### **SGP\_DISCTIM=**

Purpose: Specifies the data set disconnect time threshold in .1 millisecond increments.

Syntax: SGP\_DISCTIM{=|>|<}nnnnn

Default: None

#### **SGP\_DP@BUSY=**

Purpose: Specifies director port busy percentage to be included or excluded.

Syntax: SGP\_DP@BUSY{=|>|<}*nn*

Default: None

#### **SGP\_DPBSYDL=**

Purpose: Specifies the director port busy delay time threshold in .1 millisecond increments.

Syntax: SGP\_DPBSYDL{=|>|<}*nnnnnn*

Default: None

#### **SGP\_DVBSYDL=**

Purpose: Specifies the device busy delay time threshold in .1 millisecond increments.

Syntax: SGP\_DVBSYDL{=|>|<}*nnnnnn*

Default: None

#### **SGP\_ECMCFBS=**

Purpose: Specifies the ECAM channel programs bypassed due to busy configuration in tenths of a MB.

Syntax: SGP\_ECMCFBS=*nnnnnnnnnn*

where *nnnnnnnnnn* is 1-8 numbers.

Default: None

#### **SGP\_ECMMSGSGS=**

Purpose: Specifies ECAM messages processed in tenths of a MB.

Syntax: SGP\_ECMMSGSGS=*nnnnnnnnnn*

where *nnnnnnnnnn* is 1-8 numbers.

Default: None

#### **SGP\_ECMNSPC=**

Purpose: Specifies the ECAM channels programs bypassed due to no buffer space in tenths of a MB.

Syntax: SGP\_ECMNSPC=*nnnnnnnnnn*

where *nnnnnnnn* is 1-8 numbers.

Default: None

#### **SGP\_ECMPGMS=**

Purpose: Specifies the ECAM channel programs in tenths of a MB.

Syntax: **SGP\_ECMPGMS=nnnnnnnn**

where *nnnnnnnn* is 1-8 numbers.

Default: None

#### **SGP\_FSCBYRD=**

Purpose: Specifies the collected free space bytes read in tenths of a MB.

Syntax: **SGP\_FSCBYRD=nnnnnnnn**

where *nnnnnnnn* is 1-8 numbers.

Default: None

#### **SGP\_FSCPERC=**

Purpose: Specifies the percentage of collected free space in tenths of a percent.

Syntax: **SGP\_FSCBYRD=nnnn**

where *nnnn* is 1-4 numbers.

Default: None

#### **SGP\_FSUPERC=**

Purpose: Specifies the percentage of uncollected free space in tenths of a percent.

Syntax: **SGP\_FSUPERC=nnnn**

where *nnnn* is 1-4 numbers.

Default: None

#### **SGP\_IOPRSEC=**

Purpose: Specifies number of I/Os per-second threshold

Syntax: SGP\_IOPRSEC{=|<|>}*nnn*

Default: None

#### **SGP\_IOSQTIM=**

Purpose: Specifies the data set IOSQ time threshold in .1 millisecond increments.

Syntax: SGP\_IOSQTIM{=|>|<}*nnnnn*

Default: None

#### **SGP\_LCU@BSY=**

Purpose: Specifies LCU busy percentage to be included or excluded.

Syntax: SGP\_LCU@BUSY{=|>|<}*nnn*

Default: None

#### **SGP\_LCUID=**

Purpose: Specifies the logical control unit ID of those controllers to be included or excluded.

Syntax: SGP\_LCUID{=|>|<}*lcu ID*

Default: None

#### **SGP\_NCLPERC=**

Purpose: Specifies the net capacity load percentage in tenths of a percent.

Syntax: SGP\_FSUNPERC=*nnnn*  
where *nnnn* is 1-4 numbers.

Default: None

#### **SGP\_NRDHIT@=**

Purpose: Specifies percentage of normal reads satisfied by cache threshold.

Syntax: SGP\_NRDHIT@{=|>|<}*nnn*  
where *nnn* is 0–100.

Default: None

**SGP\_NRDPSec=**

Purpose: Specifies number of normal reads per-second threshold.

Syntax: SGP\_NRDPSec{=|>|<}*nnn*

Default: None

**SGP\_NWRHIT@=**

Purpose: Specifies percentage of normal writes satisfied by cache threshold.

Syntax: SGP\_NWRHIT@ {=|>|<}*nnn*  
where *nnn* is 0–100.

Default: None

**SGP\_NWRTPSC=**

Purpose: Specifies number of normal writes per-second threshold.

Syntax: SGP\_NWRTPSC{=|>|<}*nnn*

Default: None

**SGP\_PENDTIM=**

Purpose: Specifies the data set pending time threshold in .1 millisecond increments.

Syntax: SGP\_PENDTIM{=|>|<}*nnnnn*

Default: None

**SGP\_RDHIT@=**

Purpose: Specifies percentage of reads satisfied by cache threshold.

Syntax: SGP\_RDHIT@ {=|>|<}*nnn*  
where *nnn* is 0–100.

Default: None

**SGP\_RDSPRSC=**

Purpose: Specifies number of reads per-second threshold.

Syntax: SGP\_RDSPRSC{=|>|<}*nnn*

Default: None

#### **SGP\_READ@=**

Purpose: Specifies the percentage of IOs that are reads threshold.

Syntax: SGP\_READ@{=|>|<}*nnn*  
where *nnn* is 0–100.

Default: None

#### **SGP\_RESERV@=**

Purpose: Specifies percentage volume is reserved for inclusion or exclusion.

Syntax: SGP\_RESERV@{=|>|<}*nn*

Default: None

#### **SGP\_RESPTIM=**

Purpose: Specifies the data set response time threshold in .1 millisecond increments.

Syntax: SGP\_RESPTIM{=|>|<}*nnnnnn*

Default: None

#### **SGP\_RSFNAME=**

Purpose: Specifies the IXPF subsystem frame name.

Syntax: SGP\_RSFNAME=*xxxxxxxx*  
where *xxxxxxxx* is 1-8 characters.

Default: None

#### **SGP\_SRDHIT@=**

Purpose: Specifies percentage of sequential reads satisfied by cache threshold.

Syntax: SGP\_SRDHIT@{=|>|<}*nnn*  
where *nnn* is 0–100.



Default: None

#### **SGP\_SRDPRSC=**

Purpose: Specifies number of sequential reads per-second threshold.

Syntax: `SGP_SRDPRSC{=|>|<}nnn`

Default: None

#### **SGP\_SWRHIT@=**

Purpose: Specifies percentage of sequential writes satisfied by cache threshold.

Syntax: `SGP_SWRHIT@{=|>|<}nnn`

where *nnn* is 0–100.

Default: None

#### **SGP\_SWRPRSC=**

Purpose: Specifies number of sequential writes reads per-second threshold.

Syntax: `SGP_SWRPRSC{=|>|<}nnn`

Default: None

#### **SGP\_WRHIT@=**

Purpose: Specifies percentage of writes satisfied by cache threshold.

Syntax: `SGP_WRHIT@{=|>|<}nnn`

where *nnn* is 0–100.

Default: None

#### **SGP\_WRITE@=**

Purpose: Specifies percentage of IOs that are writes threshold.

Syntax: `SGP_WRITE@{=|>|<}nnn`

where *nnn* is 0–100.

Default: None

**SGP\_WRPSEC=**

Purpose: Specifies number of writes per-second threshold.

Syntax: SGP\_WRPSEC{=|>|<)nnn

Default: None

**SIZE=**

Purpose: Contains the size of either the primary extent or of the primary plus two secondary extents.

Allowed in: INC and EXC

Syntax: SIZE=<>nnnnnnnnnnK,M,G,T

where *nnnnnnnnnn* is 1–10 digits and K,M,G,T specifies whether the number is expressed in kilobytes, megabytes, gigabytes, or terabytes. The comparison operator symbol can be equals (=), greater than (>), or less than (<). The maximum allowable specification for SIZE is 2147483647K.

**Note:** The setting of the SIZEISPRIM global parameter affects the value associated with the SIZE parameter.

**SMF=**

Purpose: SMF is the FLST SET statement parameter that specifies the SMF message generation option for resources that are selected by the following parameters. Informational and error messages can be written to the SMF data set, or all messages can be omitted from the SMF data set. Note that the SMF option on the function definition in the SMFUNCxx member is overridden by this option for specific selected resources.

**Note:** The SMF parameter is used only for the HSM collector, Allocation, and DMS2HSM.

Syntax: MSG=[I | W | E | S | N]

I	Informational and error messages
W	Warning messages
E	Error messages only
S	Severe messages
N	No messages

Required: No

Default: None

**SMS=**

Purpose:                   Synonym. See SMSMANAGED.

**SMSMANAGED=**

Purpose:                   Specifies whether the resource is managed by DFSMS.

Allowed in:              INC and EXC and rule SET parameter for function SMSACSTE

Syntax:                  SMSMANAGED= Y/N

**SMSPOOL=**

Purpose:                   Specifies 1 to 15 SMSPOOLS that are to be used to limit volume selection during DADSM ALLOCATE for SMS-managed data sets.

Allowed in:              Rule SET parameter for function SMSSELCT

Syntax:                  SMSPOOL=(xxxxxxxx,xxxxxxxx,...)

where xxxxxxxx is an SMSPOOL.

Default:                 None

**Note:** The SMSPOOLS must contain a subset of volumes from the SMS STORGRP assigned to the data set, or the allocation will fail. In addition, if multiple pools are coded in the parameter, no attempt is made to *select* volumes from the pools in any order. The first volume that matches a volume in any of the pools will be passed.

## **SMSPOOL\_EXT=**

**Purpose:** Specifies 1 to 15 SMSPOOLS that are to be used to limit volume selection during DADSM EXTENDNV (extend to a new volume) for SMS-managed data sets.

**Allowed in:** Rule SET parameter for function SMSSELECT

**Syntax:** SMSPOOL\_EXT=(xxxxxxxx,xxxxxxxx,...)

where xxxxxxxx is an SMSPOOL.

**Default:** None

**Note:** The SMSPOOLS must contain a subset of volumes from the SMS STORGRP assigned to the data set, or the allocation will fail. In addition, if multiple pools are coded in the parameter, no attempt is made to *select* volumes from the pools in any order. The first volume that matches a volume in any of the pools will be passed.

## **SOLUTION=**

**Purpose:** Contains the solution value from the originating AUTO function command

It is recommended that each SET statement in the AUTO function include a unique solution value. This solution value can then be used on the AUTO function console command to invoke the solution.

**Allowed in:** INC and EXC in the AUTO functions

**Syntax:** SOLUTION=xxxxxxxx

where xxxxxxxx is a 1-8 character solution value

**SORT=**

**Purpose:** Specifies INC and EXC fields to sort the result group of record prior to taking any actions on the group. This can be used, in conjunction with ACT\_COUNT to take actions on the pools with the highest or lowest values in any of the pool space information fields. For example: SORT=(AP\_PERUSED,D) along with ACT\_COUNT=5 causes any specified action to be taken on the 5 highest utilized pools.

**Allowed in:** Rule SET parameter for AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.

**Syntax:** SORT=(*fieldname*,*x*,*fldname*,*x*,...) where *fieldname* is a valid INC and EXC field for the function, and *x* is either A (ascending) or D (descending)

**Default:** None. If not specified the result group of pools is not sorted and actions are taken in the order the pool information was collected.

**SPACPRIM=**

**Purpose:** Specifies the lower limit and decrement of space reduction, both as a percentage of the original primary allocation value. The first value specifies a lower limit, below which SPACPRIM will not go. The second value is the percentage by which the primary allocation will be decremented. For example, if SPACPRIM=(50,15), the function will decrement the original primary value by 15 percent on each attempt to find a primary extent but will not decrement the primary size more than 50 percent of the original value.

**Allowed in:** Rule SET parameter for function SPACPRIM

**Syntax:** SPACPRIM=(*nn*,*nn*)

where *nn* is a percentage in a range 0–90. The first value specified is the lower limit; the second value specified is the decrement amount.

If you only specify one number, it refers to the lower limit (the first number) and the decrement percentage (the second number) defaults to 10%.

**SPACSECA=**

**Purpose:** Specifies the size of the secondary space allocation as a percentage of the primary space allocation for data sets with no specified secondary allocation. For example, if the primary space allocation is 10 cylinders and SPACSECA=70, a data set with no secondary allocation specified is given 7 cylinders by SPACSECA. Also see the SPACPRIM and SPACSECI parameters.

**Allowed in:** Rule SET parameter for function SPACSECA

**Syntax:** SPACSECA=*nnn*

where *nnn* is a number in the range 1–999

**SPACSECB=**

**Purpose:** Specifies the lower limit for the space reduction as a percentage of the original secondary allocation request.

**Allowed in:** Rule SET parameter for function SPACSECB

**Syntax:** SPACSECB=*nnn*

where *nnn* is a number in the range 0–100

**SPACSECI=**

**Purpose:** Specifies the point in secondary extent processing for physical sequential data sets that the SPACSECI function automatically increases the size of the secondary allocation request. After the specified number of secondary extents have been allocated, SPACSECI increases the size of the secondary allocation by 100 percent of the original secondary allocation for each subsequent allocation. See the SPACSECI function description for an example. Also see the SPACPRIM and SPACSECA parameters.

**Allowed in:** Rule SET parameter for function SPACSECI

**Syntax:** SPACSECI=*nn*

where *nn* is a number in the range 1–15.

**SPACSECR=**

**Purpose:** Specifies the lower limit and the decrement of space reduction, both as a percentage of the original secondary allocation value. The first value specifies a lower limit, below which SPACSECR will not go. The second value is the percentage by which the secondary allocation will be decremented. For example, if SPACSECR=(50,10), the function will decrement the original secondary value by 10 percent on each attempt to find a secondary extent but will not decrement the secondary size more than 50 percent of the original value. The decrement percentage is used only for striped data sets with multiple stripes.

**Allowed in:** Rule SET parameter for function SPACSECR

**Syntax:** SPACSECR=(*nnn*,*nnn*)

where the first *nnn* is the floor limit and the second *nnn* is a percentage from 0–100 by which reduction can take place until either it fits or the floor limit is reached. (A specification of 100 will not reduce the secondary size at all.)

**Default:** SPACSECR=(0,10)

**SPACSWIR=**

**Purpose:** Specifies the lower limit and decrement amount for space reduction when adding a new volume. Both are specified as a percentage of the original primary allocation value. The first value specifies a lower limit, below which SPACSWIR will not go. The second value is the percentage by which the primary allocation will be decremented. For example, if SPACSWIR=(50,10), the function will decrement the original primary value by 10 percent on each attempt to find a primary extent but will not decrement the primary size to less than 50 percent of the original value.

**Allowed in:** Rule SET parameter for function SPACSWIR

**Syntax:** SPACSWIR=(*nnn*,*nnn*)

where the first *nnn* is the floor limit and the second *nnn* is a percentage from 0–100 by which reduction can take place until either it fits or the floor limit is reached. (A specification of 100 will not reduce the secondary size at all.)

**Default:** SPACSWIR=(0,10)

**SPACVOLA=**

- Purpose:** Specifies the maximum number of volumes on which a data set can be allocated during secondary extent processing. The SPACVOLA function adds volumes to a data set allocation, up to the limit. SPACVOLA does not support SAS data libraries. (SAS does not support OS/390 multivolume data sets.) Also see the SPACPRIM, SPACSECA, and SPACSECI parameters.
- Allowed in:** Rule SET parameter for function SPACVOLA
- Syntax:** SPACVOLA=*nn*  
where *nn* is a number in the range 1–59

**SPECIFIC=**

- Purpose:** Specifies whether a specific volume was requested for a new data set allocation.
- Allowed in:** INC and EXC
- Syntax:** SPECIFIC=Y/N

**SPLIT=**

- Purpose:** Specifies whether affinity separation should be applied to resources selected for the STKSUPP function. When unit affinity directs multiple DD statements to a single drive, STKSUPP can apply affinity separation to allocate separate devices for silo and non-silo volumes, so volumes would not have to be entered into or removed from silos.
- Allowed in:** Rule SET parameter for function STKSUPP.
- Syntax:** SPLIT=Y/N

**SQTY=**

- Purpose:** Specifies the size in kilobytes for the secondary space allocation.
- Allowed in:** Rule SET parameter for function SPACSQTY
- Syntax:** SQTY=*nnnnnK,B,G,T*  
where *nnnnn* is a number in the range 1–99999K.



**STEP=**

**Purpose:** Contains the stepname of a jobstep. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** STEP=xxxxxxx

where xxxxxxx is a valid jobstep name 1–8 characters long

**STEPACCT<sub>n</sub>=**

**Purpose:** Contains the *n*th subfield in the ACCT field of the EXEC JCL statement. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** STEPACCT<sub>n</sub>=xxxxxxx

where *n* is a number in the range 1–3 and xxxxxxx is a character string 1–20 characters long

**STOGROUP=**

**Purpose:** Specifies or contains the DFSMS storage group name for a data set. MAINVIEW SRM name masking can be used for filter list entries. Rule list entries must specify a valid DFSMS storage group name.

**Allowed in:** INC and EXC and rule SET parameter for function SMSACSSC

**Syntax:** STOGROUP=xxxxxxx

where xxxxxxx is a valid storage group name 1–8 characters long.

**STORCLAS=**

**Purpose:** Specifies or contains the DFSMS storage class name for a data set. MAINVIEW SRM name masking can be used for filter list entries. Rule list entries must specify a valid storage class name.

**Allowed in:** INC and EXC and rule SET parameter for function SMSACSSC

**Syntax:** STORCLAS=xxxxxxx

where xxxxxxx is a valid storage class name 1–8 characters long.

**STORGRP=**

Purpose:                   Synonym (See STOGROUP)

**STRIPCNT=**

Purpose:                   Contains the number of stripes allocated to the data set. The STRIPCNT parameter is used to determine the number of stripes that are allocated to an extended format sequential data set. This parameter will always be one for a single stripe data set.

Allowed in:             INC and EXC

Syntax:                 STRIPCNT=*nnnnnnnn*

where *nnnnnnnn* is a number in the range 1–99999999

**Note:** Not valid during SPACPRIM processing. The current OS/390 DFP maximum is 16 stripes.

**STRIPTY=**

Purpose:                   Specifies the type of extended format data set.

Allowed in:             INC and EXC

Syntax:                 STRIPTY=*xx*

where *xx* is a value from the following list:

SS	Single Stripe SAM data sets
SM	Multi-Stripe SAM data sets
VS	Single Stripe VSAM data sets

**Note:** Not valid during SPACPRIM processing.

**SUPVOL=**

Purpose:                   Specifies that allocation requests for specific volumes are suppressed (SUPVOL=Y) or allowed (SUPVOL=N). Suppression of specific volume requests allows MAINVIEW SRM to allocate to any volumes in the eligible pools. Also see the parameters USEVOL and MNTYPE.

Allowed in:             Rule SET parameter for function DASDPOOL

Syntax:                 SUPVOL=Y/N

Default                 SUPVOL=Y

**SYSID=**

**Purpose:** Contains an OS/390 operating system identifier. MAINVIEW SRM name masking can be used.

**Note:** This parameter is not available for functions DSNCHECK, SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, or SMSACSTE.

**Allowed in:** INC and EXC

**Syntax:** SYSID=xxxx

where xxxx is a system identifier 1–4 characters long

**TEMPDSN=**

**Purpose:** Flags temporary data sets. This parameter is unique to the application collector.

**Allowed in:** INC and EXC

**Syntax:** TEMPDSN=Y/N

**TRKCYL=**

**Purpose:** Specifies the number of tracks per cylinder of the source volume for the SPACCONV function. The value is used to calculate the proper size of an allocation on new devices for specifications based on devices no longer used.

**Allowed in:** Rule SET parameter for function SPACCONV

**Syntax:** TRKCYL=nn

where nn is a number in the range of 1–99.

**TRKLEN=**

**Purpose:** Specifies the number of bytes per track of the source volume for the SPACCONV function. The value is used to calculate the proper size of an allocation on new devices for specifications based on devices no longer used.

**Allowed in:** Rule SET parameter for function SPACCONV

**Syntax:** TRKLEN=nnnnn

where nnnnn is a number in the range 1–99999.

## UNIT=

**Purpose:** Contains or specifies the unit generic name (esoteric or generic) to be used by the data set during allocation.

**Note:** To use the UNIT= filter on VTS data sets, you must specify JCLEXT=NO in the SMMSYSxx member.

**Allowed in:** INC and EXC and the rule SET parameters for functions DASDPOOL, SPACVOLA, and TAPECOMP

**Syntax:** UNIT=xxxxxxxx

where xxxxxxxx is a valid unit name (esoteric or generic) 1–8 characters long.

## USECPOOL=

**Purpose:** Specifies whether the current pool or the alternate pool is searched to find an additional volume required by a secondary allocation. USECPOOL=Y specifies that the current pool is searched, followed by the alternate pool, if any. USECPOOL=N specifies that the current pool is not searched; the search begins with the alternate pool, if any. If USECPOOL=N and no alternate pool is specified, the secondary allocation attempt fails. Also see the parameters SPACVOLA, MNTYPE, and ALTPool.

**Allowed in:** Rule SET parameter for function SPACVOLA

**Syntax:** USECPOOL=Y/N

**Default:** USECPOOL=Y

## USER=

**Purpose:** Contains the user name (RACF or CA-Top Secret; for CA-ACF2, contains the logon ID). MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** USER=xxxxxxxx

where xxxxxxxx is a valid user name 1–8 characters long.

**USEVOL=**

Purpose:	Specifies the type of volume (storage, private, or all) that satisfies non-specific allocation requests. Also see the SUPVOL and MNTYPE parameters.
Allowed in:	Rule SET parameter for functions DASDPOOL, FDRASIST, TAPEPOOL, and HSMRECAL
Syntax:	USEVOL=xxxx  where <i>xxxx</i> is a value from the following list:  STOR Storage mounted volume PRIV Privately mounted volume ALL Volume of any mount type
Default:	ALL

**USRCn=**

Purpose:	Character field for a user-specified variable 1–8 characters long. The value of <i>n</i> can be 1–10 (for example USRC1, USRC2, and so forth).
Allowed in:	INC and EXC parameters and the rule SET parameter for function USERVARS
Syntax:	USRCn=xxxxxxxxxx

**USRNy=**

Purpose:	Numeric field for a user-specified variable. The value of <i>y</i> can be 1–10 (for example USRN1, USRN2, and so forth)
Allowed in:	INC and EXC parameters and the rule SET parameter for function USERVARS
Syntax:	USRNy=nnnnnnnn  where <i>nnnnnnnn</i> does not exceed 214783647

**VCOMPLLQ=**

**Purpose:** Specifies or contains the low-level qualifier of a VSAM data set component. MAINVIEW SRM name masking can be used for filter list entries.

**Allowed in:** INC and EXC and rule SET parameter for function DSNCHECK

**Syntax:** VCOMPLLQ=xxxxxxx

where xxxxxxx is a valid data set name qualifier 1–8 characters long. It will contain blanks for the cluster.

**VFORCE=**

**Purpose:** Specifies that naming conventions for VSAM components will be forced by adding standard component suffixes (DATA, INDEX) to VSAM data set cluster names.

**Allowed in:** Rule SET parameter for function DASDPOOL

**Syntax:** VFORCE=Y/N

Y Overrides unspecified or invalid VSAM component names by appending .DATA and .INDEX qualifiers to the cluster name.

N Default

**VIO=**

**Purpose:** Specifies that a data set should be allocated in main storage (VIO=Y) or on DASD (VIO=N).

**Allowed in:** Rule SET parameter for function VIOALLOC

**Syntax:** VIO=Y/N

**VOL=**

**Purpose:** Contains the volume serial number. MAINVIEW SRM name masking can be used.

**Allowed in:** INC and EXC

**Syntax:** VOL=xxxxxx

where xxxxxx is a valid volume serial number 1–6 characters long.

**VOLSEL=**

**Purpose:** Specifies the method of volume selection from a pool.

**Allowed in:** Rule SET parameter for function DASDPOOL, FDRASIST, and SMSSELECT.

**Syntax:** *VOLSEL=BESTFIT/CRITDSN/DPO/HISTDPO/MAXSPACE/  
PERCENT*

**BESTFIT** - Specifies that the volume with the smallest contiguous extent that satisfies the primary allocation should be selected.

**CRITDSN** - Controls allocations by specifying data sets that should not reside on the same volume.

**Warning!** CRITDSN is resource intensive and should be used only for a small list of critical data sets. It should not be used without considering the system impact.

**DPO** - Specifies volume selection based on performance statistics accumulated by MAINVIEW SRM.

**HISTDPO** - Specifies volume selection based on historical volume performance.

**Warning!** HISTDPO requires more resources than normal allocations.

**MAXSPACE** - Specifies that the volume with the largest single contiguous extent should be selected.

**PERCENT** - Specifies that the volume with the largest amount of free space should be selected.

**VOLSER=**

**Purpose:** Specifies the volume serial ID or accepts a special asterisk mask in which leading asterisks require the new volume name to match the existing volume name in the leading asterisk positions. For example, VOLSER=(\*\*\*) will add only volumes for which the first three characters match the existing volume.

Also, for compatibility with STOP-X37 comparison, triplets can be specified with the first operand in the triplet specifying a partial volume name, the second operand specifying the offset into the volume name to start the comparison, and the third operand is the comparison operator. For example, VOLSER=((WRK,1,EQ),(PROD,1,EQ)) will allow SPACVOLA to add volumes that start with the characters WRK or PROD. Valid operators are

EQ	=
GT	>
LE	<=
NE	≠
LT	<
GE	>=

**Allowed in:** Rule SET parameter for function SPACVOLA

**Syntax:** VOLSER=xxxxxx or VOLSER=(\*\*\*\*\*)

where xxxxxx is the 1–6 volume serial ID or (\*\*\*\*\*) is 1–5 asterisks

**VSAMCOMP=**

**Purpose:** Contains the VSAM data set component type (DATA or INDEX).

**Allowed in:** INC and EXC

**Syntax:** VSAMCOMP=xxxxx

where xxxxx is a value from the following list:

DATA	Data component of data set
INDEX	Index component of data set



**VSAMDEF=**

Purpose: Contains the VSAM data set cluster definition (DATA or INDEX).

Allowed in: INC and EXC

Syntax: VSAMDEF=xxxxxxx

where xxxxxx is a value from the following list:

CLUSTER	The data or index component of a base cluster
AIX	The data or index component of an alternate index that is not part of an upgrade set
UPGRADE	The data or index component of an alternate index that is part of an upgrade set

**VSAMSEP=**

Purpose: Contains an indicator of whether data and index components are on separate volumes.

Allowed in: INC and EXC

Syntax: VSAMSEP= Y/N

**XMODE=**

Purpose: Contains the execution mode of a job.

Allowed in: INC and EXC

Syntax: XMODE=xxx

where xxx is a value from the following list:

STC	Started task
TSO	TSO session
JOB	Batch job



---

# Appendix F Troubleshooting

This appendix provides a list of frequently asked questions developed by BMC Customer Support for MAINVIEW SRM.

Table F-1 Frequently Asked Questions (Part 1 of 4)

Question/Problem	Answer/Resolution
<b>Compatibility</b> Does MAINVIEW SRM support devices HITACHI 7700 AND 9960?	MAINVIEW SRM Allocation and Reporting support HITACHI devices. MAINVIEW SRM does not provide reporting beyond standard IBM available statistics/ information. That is, MAINVIEW SRM does not provide special RAID information (RAID director, PAV, controller statistics, and so on).
<b>IGD17101I</b> If NOCATLG2 is work correctly, why am I receiving the IBM message: IGD17101I DATA SET <i>dsname</i> NOT DEFINED BECAUSE DUPLICATE NAME EXISTS IN CATALOG RETURN CODE IS <i>rc</i> REASON CODE IS <i>rsnc</i> IGG0CLxx	<p>Message IGD17101I stating that a “duplicate data set exists” can still be issued even if NOCATLG2 processing is successful. The IGD17101I message is even issued after MAINVIEW SRM issues the successful recatalog message. This occurs because MAINVIEW SRM intercepts SMS processing and then drives SMS VTOC/data set services (VDSS) from the code to determine if each catalog/ allocation process is going to be successful. At this time, if the data set being allocated <i>new</i> already exists, SMS will queue the IGD17101I message, assuming that the system is returning to the module that will display the message. Instead, the system returns to MAINVIEW SRM and detects the allocation error, performs NOCATLG2 processing, and redrives VDSS for a second attempt.</p> <p>During NOCATLG2 processing and before redriving VDSS, MAINVIEW SRM issues messages. MAINVIEW SRM then redrives VDSS and the successful allocation message is queued behind the IGD17101I message. When MAINVIEW SRM then regains control and, if all allocations are successful, returns control to SMS. Then SMS issues both queued messages. As in the NOCATLG2 case, the successful allocation message, (IGD101I) will also appear if the allocation is eventually successful.</p>
<b>Security</b> We do not want to use the security but it defaults in the log. Error message BBMXCL16E.	<p>To prevent the error message, add the following DD statement to your SVOS Startup PROC:</p> <p>//BBSECURE DD DISP=SHR,DSN=xxxx.xxxxx.xxxxx.BBACTDEF</p>

**Table F-1**      **Frequently Asked Questions (Part 2 of 4)**

Question/Problem	Answer/Resolution
<b>SG-Auto/Sysplex</b> Can SG-Auto be run in multiple LPARs? The following message appears at startup: SGA228I - Start of task already active and SVM0155I - Token not found. Shut down continue... Trying to start product on multiple systems...	SMMSYSxx parameter, SGA_ENQSCOP, defaults to GLOBAL. When SG-Auto starts up in one LPAR of a GRS system with the parameter set to GLOBAL, an ENQ is placed on its resource across systems. If SG-Auto tries to start up in another LPAR with the SMMSYSxx parameter also set to GLOBAL, it finds the resource has already been enqueued. To run SG-Auto in multiple LPARs in the same GRS system, only one SMMSYSxx SGA_ENQSCOP can be set to GLOBAL. The others have to be set to LOCAL. Or, the parameter in all SMMSYSxx members can be set to LOCAL.
Getting: "ADDRESS TSO XCMD +++ RC(-3) +++"	The BUDDSN command must run authorized. One of the installation steps is to add BUDDSN (and BUDGET) to the authorized command table in SYS1.PARMLIB. If this is not done, no function involving BUDDSN will work. (The REXX procedure SGCDNL, listed in the ISPF trace, uses BUDDSN.) The following is a scenario to verify that BUDDSN is working properly: 1. Stop ISPF, returning to the READY prompt. 2. Issue the command TSOLIB ACTIVATE DSN(---name-of-BBLINK--) 3. Restart ISPF and re-test BUDDSN. BMC Software does support TSOLIB. Just as with STEPLIB, TSOLIB provides support for APF authorized libraries. BMC Software cannot guarantee, nor validate, that some other type of dynamic "STEPLIB" function provides the same degree of service.
<b>Shutdown</b> Bringing down SVOS started task generated the following message: MSG WAITING TO BECOME DORMANT. What should I do?	If you are not going to IPL at this time, you should reply "blank" to this message. The product is waiting to resolve ENQUEUEs. You should display RES=ETILOAD for ENQUEUE. Review all jobs that show ETILOAD ENQUEUE. Resolve the ENQUEUEs and the product will shut down. You can then restart with no problem.
<b>Startup</b> Why does SYS1.LPALIB have to be on the DD within the PROC?	The modules have to be available at startup, either in the PROC or the SMMSYSxx member, in order for the system to checks that the modules in the LPA library match and to ensure there is not a mismatch in the entry points for the hooks. It determines where the entry points should be for the hooks and then goes to IBM modules in core to make sure that MAINVIEW SRM are putting the hooks in the correct place. Therefore, it has to be available either at startup with the ENQ or to be checked and the ENQ freed in the SMMSYSxx member.
<b>Startup</b> Why is a DD statement for SYS1.LPALIB required within SVOS?	The DD statement can be coded in your SMMSYS00 global member as: SYSLIB=SYS1.LPALIB. When it was in the SVOS JCL, an enqueue occurred that required bringing down the started task (SVOS) to update the LPALIB. By using the SYS1.LPALIB parameter, the enqueue is freed.
Do PTFs that address EMC API only bypass the error messages or do they handle the new MicroCode Levels?	Currently BMC Software supports EMC Symm API Release 4.1.0, the current EMC API at all current microcode levels. These levels are not documented well in the EMC manuals because the numbers change before the books can be produced. Usually, the MicroCode Levels (MCLs) can be found in the EMC macro calls, but even these have been found to be incomplete. When an MCL is too low or invalid or so new that it is not support, the MAINVIEW SRM code displays an appropriate error message. In the event of a higher MCL (new), MAINVIEW SRM displays the box number along with the unsupported MCL in the message. Data for that box is dropped from the RAID DATA TABLE and the process continues with the next box.

**Table F-1** Frequently Asked Questions (Part 3 of 4)

Question/Problem	Answer/Resolution
How can we set the scale to gigabytes instead of bytes in the Space Utilization batch report?	The way to affect the scale on this view is to use the CUST command and alter the field sizes. The CUST command is documented in the <i>Using MAINVIEW</i> manual. The product does not give you the ability to actually set the scale.
<b>SVM0036E</b> MEMBER IN ERROR: SMMSYS00 <b>SVO2121W</b> UNABLE TO CONNECT TO BBI3, NO BBI3_SSID IN SMMSYS	Things to CHECK:  Verify that you are not receiving message: *SVM1002W **WARN** THE LICENSE FOR STOPX37/II HAS EXPIRED SVM0090E UNAUTHORIZED ATTEMPT TO USE SVOS  <ul style="list-style-type: none"> <li>• If there no valid password in the SMMSYSxx member, you will also receive the BBI3_SSID error message. If you add a valid password, the BBI3_SSID error will go away.</li> <li>• If the error remains, review the SSID you placed in the CAS and verify that is what you have specified in you SMMSYSxx for BBI3_SSID.</li> </ul>
<b>SVM4001I</b> message occurs at the same time as the IEC070I 104-204 bearing the same DB2 data set name. StopX37/II has been told not to interfere with DB2 data sets. Why might these messages being produced? Are there any zap/PTFs to address this problem?	The product is performing as documented.  'SVM4001I Recovery Attempt Failed to Pass FLST/RLST Criteria' This informational message is telling you that your code has chosen not to provide any service for this DB2 DSN. If you do not want to see this message, you will have to alter the level of messages you are having the product produce. You can go to the FLST for the functions in question and change/add a message parameter as follows: SET MODE=INACT MSG=E INC DSNTYPE = DB2 SET MODE = ACT, MSG=I INC xxxxx
<b>SVO0006E</b> : There was an error in the startup parameters for the svos component <b>SVO0006E</b> : A successful system-level refresh (SMMSYSxx) must be done before this component can be started	To identify the system member in error, issue the following console command:  F SVOS, SVOS R,SYS=xx  If the error is the BBI3_SSID parameter, update it in the SMMSYSxx member. To update the value of BBI3_SSID, SVOS must be stopped and restarted; it cannot be refreshed. The CAS subsystem name is specified in the SSID= parameter on the PARM= keyword for the CAS JCL EXEC statement.  <b>Note:</b> Even if you are not using the MAINVIEW panels, you must have a value in the BBI3_SSID parameter in the SMMSYSxx to start SVOS.

**Table F-1**      **Frequently Asked Questions (Part 4 of 4)**

Question/Problem	Answer/Resolution
<p><b>SVO0612</b> START FOR <i>prd_name</i> FAILED ON <i>subsys</i>  Explanation: SVOS could not start the product (<i>prd_name</i>). Subsys is the subsystem name used by this instance of SVOS.  Message Level: None  User Response: A previous SVOS message should explain why the <i>prd_name</i> could not be activated.</p>	<p>SVALLOC and a product called xxxxx were started at the same time. Both products are trying to hook IBM allocation modules at the same time. You should delay the start of one product by a couple seconds, which will stop this from happening.</p>
<p><b>Sysplex</b>  When running MAINVIEW SRM performance collectors in sysplex with multiple LPARs, does each LPAR need its own file?</p>	<p>There can be up to 99 different performance collector databases allocated and viewed within MAINVIEW SRM.</p>
<p><b>Sysplex</b>  Are there data sets that need to be shared between all SYSPLEXs?</p>	<p>The only data sets that we suggest that they share across LPARS is the Application databases. The other databases can, and probably should be system specific. What you will want to think about is if the LPARS within the sysplexes use the same DASD configurations they might not want or need to run a performance collector on each system (since it will be collecting and reporting the same information). You will, on the other hand, want to run individual performance collectors because the system-performance statistics might differ per system.</p>
<p><b>Sysplex</b>  What data sets must be unique for each sysplex?</p> <p>If we make these data sets unique for each SYSPLEX, will we be able to view all SYSPLEXs / LPARs from any one system running MAINVIEW?</p>	<p>SVSGP must run as a started task on each LPAR and each LPAR must have a unique performance collector database file.  The information that the performance collector collects is LPAR specific and must run in each LPAR with its own unique database.</p> <p>You will be able to view information for each LPAR within the sysplex that you are on. You will not be able to view PRSM data from the PPF sysplex.</p> <p>As for sysplex reporting, we can show data from any LPAR that is in the same VTAM network as the local LPAR. What sysplex the LPAR is in really does not matter, or whether the LPAR is in a sysplex. The key is that the CAS on the local LPAR has to be connected with a CAS on the remote LPAR though VTAM definitions you give to the CAS.</p>
<p><b>Sysplex</b>  Should the UBBPARM, UBBSAMP, and UBBPROC only contain <i>only</i> the members we have modified?</p>	<p>UBBPARM, UBBSAMP, and UBBPROC contain site-modified parameter members. Also, you can have a separate BBPARM/UBBPARM per sysplex, per LPAR, or you can have one BBPARM/UBBPARM that shared among all sysplexs, all LPARS in a sysplex, or any other combination. The point is that you can use one BBPARM/UBBPARM for all LPARS across all sysplexs if you want to, using the FORSYSID= and/or FORSYSLEX= keywords in the PARMLIB members. The advantage is that everything is in one place.</p> <p>The way to think about the sysplexing capabilities is that it provides the customer with the ability to build and maintain one set of UBBPARM libraries (per sysplex) where all pools/globals/FLST/RSMT members can be stored. It also provides the capability to view and edit MVW data on multiple LPARS within a sysplex from a common system. To do this, you still need the same DB definitions as they currently have. Most of the changes relate to adding the global parameters to allow sysplexing to the UBBPARM members to define your environment.</p>

---

# Glossary

This glossary defines BMC Software terminology. Other dictionaries and glossaries can be used in conjunction with this glossary.

Because this glossary pertains to BMC Software MAINVIEW SRM-related products, some of the terms defined might not appear in this book.

To help you find the information you need, this glossary uses the following cross-references:

***Contrast with*** indicates a term that has a contrary or contradictory meaning.

***See*** indicates an entry that is a synonym or contains expanded information.

***See also*** indicates an entry that contains related information.

**abend** An abnormal end of a task.

## **Access Control Facility (ACF)**

A data security software package marketed by Computer Associates that controls which users are allowed to access a computer system and limits the actions they are allowed to take.

## **Access Method Services (AMS)**

In IBM, the facility used to define and reproduce VSAM key-sequenced data sets (KSDS).

**ACF2** *See* Access Control Facility (ACF).

**ACS routine** A procedural set of automatic class selection (ACS) language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

---

<b>action parameters</b>	<i>See</i> SET keywords.
<b>address space</b>	<p>A range of contiguous virtual storage addresses that the system creates for a user. It contains user data and programs, as well as system data and programs, some of which are common to all address spaces. One of three elements can run in a single address space:</p> <ul style="list-style-type: none"> <li>• batch job</li> <li>• TSO user</li> <li>• started task</li> </ul> <p>Each virtual storage address space is isolated from all others (no address space can see the storage for another address space). This isolation is accomplished through segment and page tables. Each address space has its own segment and page tables. Therefore, a single page table will only point to a real storage page frame, thus making it impossible for a separate address space to access or change that page unless the program is APF authorized.</p>
<b>allocate</b>	To assign a resource, such as a volume or disk space.
<b>AMS</b>	<i>See</i> Access Method Services (AMS).
<b>AO</b>	System term used for MAINVIEW AutoOPERATOR.
<b>APAR</b>	<i>See</i> Authorized Program Analysis Report (APAR).
<b>application</b>	In MAINVIEW SRM, a grouping of information in the application database that represents an entity (a function, a user, a department, and so forth) with space utilization that requires monitoring or control or that requires both monitoring and control. This information is related to an application code.
<b>ATL</b>	Automatic tape libraries.
<b>Authorized Program Analysis Report (APAR)</b>	
	A report of a problem caused by a suspected defect in a current unaltered release of a program. <i>See also</i> program temporary fix (PTF).
<b>AUTO commands</b>	The console commands that are used to initiate automation: AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.
<b>AUTO functions</b>	The functions that are used to define and activate storage automation: AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL.
<b>AutoCustomization</b>	The online facility that is used for customizing the installation of products. AutoCustomization provides an ISPF panel interface that both presents customization steps in sequence and provides current status information about the progress of the installation.



---

## Backup Control Data Sets (BCDS)

In DFSMSHsm, a VSAM key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMSHsm.

**BBI** *See MVI.*

## BBI subsystem product address space (BBI-SS PAS)

The OS/390 subsystem address space that manages communication between local and remote systems and that contains one or more of the following products:

- MAINVIEW AutoOPERATOR
- MAINVIEW for CICS
- MAINVIEW for DB2
- MAINVIEW for DBCTL
- MAINVIEW for IMS Online
- MAINVIEW for MQSeries (formerly Command MQ for S/390)
- MAINVIEW SRM
- MAINVIEW VistaPoint (for CICS, DB2, DBCTL, and IMS workloads)

**BBPARM** *See parameter library.*

**BBPROC** *See procedure library.*

**BBPROF** *See profile library.*

**BBSAMP** *See sample library.*

**BBXS** *See BMC Software Subsystem Services (BBXS).*

**BCDS** *See Backup Control Data Sets (BCDS).*

## BMC Software Subsystem Services (BBXS)

The common set of service routines loaded into common storage and used by several BMC Software MAINVIEW products.

**border** A visual indication of the boundaries of a window.

**Budget** The maximum amount of space (as determined by the space administrator) allotted for a given application.

**CA-1** The Computer Associates tape library management system.

**CA-ACF** *See ACF2.*

---

<b>CA-Disk</b>	The data management system by Computer Associates that replaces the DMS product.
<b>CAS</b>	<i>See</i> Coordinating Address Space (CAS).
<b>CA-Top Secret</b>	A data security software package from Computer Associates similar to RACF or ACF2 that limits access to computer resources.
<b>CDS</b>	<i>See</i> Control Data Sets (CDS).
<b>CLIST</b>	<i>See</i> Command List (CLIST).
<b>cluster</b>	In systems with VSAM, a cluster is a named structure consisting of a group of related components, for example, a data component with its index component.
<b>CMF MONITOR Extractor</b>	A component of CMF that collects performance statistics for CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390, and RMF postprocessor.
<b>CMFMON</b>	<p>A component of CMF MONITOR that simplifies online retrieval of information about system hardware and application performance and creates MAINVIEW SMF-type 79 records.</p> <p>The CMFMON online facility can be used to view data in one or more formatted screens.</p> <p>The CMFMON write facility can be used to write collected data as MAINVIEW SMF-type 79 records to an SMF or sequential data set.</p>
<b>collection interval</b>	A length of time data is collected. <i>See also</i> delta mode, total mode.
<b>column</b>	A vertical component of a view or display, typically containing fields of the same type of information, that varies by the objects associated in each row.
<b>command line</b>	A line in the control area of the display screen where primary commands can be typed. <i>Contrast with</i> line command column.
<b>Command List (CLIST)</b>	A control language used to manage interactive applications in the TSO/OS/390 environment. In TSO, CLIST is a data set or member of a partitioned data set (PDS) containing TSO commands to be performed sequentially in response to the EXEC command. It is likely to be integrated with or superseded by the REXX procedural language.

---

## comparison operators

An operator used in comparison expressions. Comparison operators valid in MAINVIEW SRM are = (equal), > (greater than), < (less than, and = (not equal to).

## Control Data Sets (CDS)

The data sets that contain information that is used to manage the functions for the task with which they are associated.

## CONTROL-T

The tape management system from BMC Software.

## Coordinating Address Space (CAS)

An address spaces that is used by the MAINVIEW window environment architecture. The CAS supplies common services and enables communication between linked systems. Each OS/390 or z/OS image requires a separate CAS. Cross-system communication is established through the CAS using VTAM and XCF communication links.

## current data

The data that reflects the system in its current or real-time state. *Contrast with* historical data and interval. *See also* real-time data.

## Customize or Customization

A method of supplying initial parameters so that a product will run.

## DADSM

*See* Direct Access Device Space Management (DADSM).

## DASD

*See* Direct Access Storage Device (DASD).

## DASD space administrator

An individual that is assigned to perform DASD storage administration functions.

## data class

A list of the data set allocation parameters and their values that are used when allocating a new SMS-managed data set in an OS/390 operating environment.

## data collector

In MAINVIEW, a program that collects data from various sources and stores the data in records used by views.

## data definition name

The name of a data definition (DD) statement that corresponds to a data control block that contains the same name.

---

## **data definition statement**

A job control statement describing a data set associated with a specific job step.

## **Data Facility Storage Management System (DFSMS)**

In IBM, an operating environment that helps automate and centralize the management of storage. To manage storage, DFSMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

## **Data Facility Storage Management System data facility product (DFSMSdfp)**

The heart of the IBM storage management subsystem. It provides the logical and physical input and output for z/OS storage, it keeps track of all data and programs managed within z/OS, and it provides data access both for native z/OS applications and other platforms such as AIX/UNIX, the Windows family, AS/400, or OS/2.

## **Data Facility Storage Management System hierarchical storage management (DFSMSHsm)**

An IBM program that facilitates the automatic movement of files from hard disk to slower, less-expensive storage media. The typical hierarchy is from magnetic disk to optical disk to tape.

## **Data Facility Storage Management System removable media manager (DFSMSrmm)**

A functional component of DFSMS for managing removable media resources including automatic libraries (such as the IBM Virtual Tape Server), for the z/OS environment.

## **Data Set Control Block (DSCB)**

A control block in the VTOC that describes data set physical characteristics such as size, organization, and so on.

## **data striping**

A technique for achieving parallel transfer of data by splitting the data across a number of devices and transferring parts of sets of data in parallel.

## **DD statement**

*See* data definition statement.

## **ddname**

*See* data definition name.

---

## Default Application Code Construction Rules

A set of rules held in the MAINVIEW SRM application database that defines the number of application codes that can be associated with a space allocation request and how construction rules are built.

**DFSMS** *See* Data Facility Storage Management System (DFSMS).

**DFSMSdfp** *See* Data Facility Storage Management System data facility (DFSMSdfp).

**DFSMSHsm** *See* Data Facility Storage Management System hierarchical storage management (DFSMSHsm).

**DFSMSrmm** *See* Data Facility Storage Management System removable media manager (DFSMSrmm).

## Direct Access Device Space Management (DADSM)

A component of MVS/DFP. The DADSM exit routines are provided by IBM to help your installation control DASD space requests and enforce installation standards.

## Direct Access Storage Device (DASD)

A device with rotating recording surfaces that provides immediate access to stored data.

**DMS2HSM** *See* MAINVIEW SRM DMS2HSM.

**DSCB** *See* Data Set Control Block (DSCB).

**dump** A representation of all or part of the contents of internal storage of a computer. The printout is often used to diagnose program errors. A dump is usually created as a result of a major program error.

**EasyHSM** *See* MAINVIEW SRM Reporting.

**EasyPOOL** *See* MAINVIEW SRM Allocation.

**EasySMS** *See* MAINVIEW SRM Allocation.

**enqueue** The OS/390 expression (abbreviated as ENQ) for requesting resource serialization. ENQ can be used to put a user-named entry in the system resource queue to prevent another program from using a particular resource. Reserves a device or resource.

**enterprise** A term used to encompass an entire business group, organization, or corporation, including all local, remote, and satellite offices.

---

## Enterprise Storage Automation

*See* MAINVIEW SRM Automation.

### Enterprise Storage Server (ESS)

The IBM Enterprise Storage Server (ESS) products, also known as Shark or 2105 Control Unit, that are designed for use in the enterprise disk storage market.

**ESS** *See* Enterprise Storage Server.

**event** A user-defined message sent to AutoOPERATOR by MAINVIEW SRM. User-defined storage occurrences can cause events to be sent to AutoOPERATOR, where actions can be initiated in response to the storage condition.

**exit routine** A user-supplied program attached to the operating system that is used to customize system functions to meet specific installation needs.

**extended VSAM** 1) extended format: The format of a data set that has a data set name type (DSNTYPE) of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different.  
  
2) extended addressability: The ability to create and access a VSAM data set that is greater than 4 GB in size. Extended addressability data sets must be allocated with DSNTYPE=EXT and EXTENDED ADDRESSABILITY=Y.

**extent** One continuous portion of space on a DASD volume consisting of empty free space, a data set, or a portion of a data set. When a data set cannot be contained in one piece of continuous free space, it must be split into pieces (extents). If it requires more space than originally requested in the primary space request in the JCL, OS/390 might ask for secondary extents. Each piece of secondary is an extent of the data set. MVS must satisfy a space request within five extents (primary or secondary) or it will issue an X37 or primary space-not-available abend. OS/390 will not allow a non-VSAM data set to be split into more than 16 extents on a single disk volume. VSAM data sets can occupy up to 123 total extents across multiple volumes.

**field** A group of character positions within a screen or report used to type or display specific information.

**field help** The online Help that describes the purpose or contents of a field on a screen. To display field Help, place the cursor anywhere in a field and press PF1 (HELP). In some products, field Help is accessible from the screen Help that is displayed when you press PF1.

---

<b>filter</b>	The selection criteria used to limit the number of rows displayed in a view. Data that does not meet the selection criteria is not displayed. A filter is composed of an element, an operator, and an operand (a number or character string). Filters can be implemented in view customization, through the PARM/QPARM commands, or through the Where/QWhere commands. Filters are established against elements of data.
<b>filter list</b>	The parameters that are used to select resources for one or more functions and define the processing mode of those resources to be active, inactive, or simulate. Multiple selection criteria can be specified in a filter list. The function definition identifies the filter list to be used. <i>See also</i> rule list.
<b>fire</b>	The term used to indicate that an event has triggered an action. In MAINVIEW AutoOPERATOR, when a rule selection criteria matches an incoming event and fires, the user-specified automation actions are performed. This process is also called handling the event.
<b>fixed field</b>	The field that remains stationary at the left margin of a screen that is scrolled either right or left.
<b>FLST</b>	<i>See</i> filter list.
<b>high-level qualifier</b>	The first node or portion of a data set name.
<b>historical data</b>	<ol style="list-style-type: none"><li>1) The data that reflects the system as it existed at the end of a past recording interval or the duration of several intervals.</li><li>2) Any data stored in the historical database and retrieved using the TIME command. <i>Contrast with</i> current data, interval data and real-time data.</li></ol>
<b>historical data set</b>	In MAINVIEW products that display historical data, the VSAM cluster file in which data is recorded at regular intervals.
<b>historical database</b>	A collection of performance data that is written at the end of each user-defined recording interval and that contains up to 100 VSAM clusters. The data is extracted from the historical database with the TIME command. <i>See</i> historical data.
<b>form</b>	One of two constituent parts of a view; the other is query. A form defines how the data is presented; a query identifies the data required for the view. <i>See also</i> query, view.

---

## Host Software Component (HSC)

A library-management software from StorageTek that manages the library and the interface between the operating system and each automated cartridge system. It maintains, in a DASD control data set, records of the cartridges stored in each automated cartridge system. The HSC intercepts mount/dismount messages, translates them into move requests and routes them to the LMU.

**HSC** *See* Host Software Component (HSC).

**HSM** *See* Data Facility Storage Management System hierarchical storage management (DFSMSHsm).

**hyperlink** A preset field in a view or an EXPAND line on a display that permits you to

- access cursor-sensitive help
- issue commands
- link to another view or display

The transfer can be either within a single product or to a related display/view in a different BMC Software product.

## Indexed Sequential Access Method (ISAM)

A method of file access in which a stored index contains the address of a group of records. Has been functionally stabilized and replaced by VSAM.

## Install or Installation

The combination of unloading and customizing. *See* unload and customize.

## Interactive System Productivity Facility (ISPF)

An IBM licensed menu and screen management system that allows the programmer to communicate with the computer system in a very simple way. Provides frequently performed functions in a streamlined fashion to help make programmers more productive. Provides interactive panel-driven (menu) dialogs. Used with TSO, ICCF, and CMS environments.

**interval data** The cumulative data collected during a collection interval. Intervals usually last from 15 to 60 (60 is the SMF interval standard) minutes depending on how the recording interval is specified during product customization. *See also* historical data.

*Contrast with* current data and real-time data.

**ISAM** *See* Indexed Sequential Access Method (ISAM).



---

**ISPF** *See Job Control Language (JCL).*

**Job Control Language (JCL)**

A problem-oriented language designed to express statements in a job that identify the job or describe its requirements to an operating system.

**LCU** *See Logical Control Unit (LCU).*

**line command** A command that you type in the line command column in a view or display. Line commands initiate actions that apply to the data displayed in that particular row.

**line command column**

The command input column on the left side of a view or display. *Contrast with* command line.

**Logical Control Unit (LCU)**

In RMF reports, the logical representation of a physical control unit or a group of physical control units with one or more devices in common.

**MAINVIEW** The BMC Software integrated systems management architecture.

**MAINVIEW Alarm Manager (MV ALARM)**

In conjunction with other MAINVIEW products, notifies you when an exception occurs. MAINVIEW Alarm Manager is capable of monitoring multiple systems simultaneously, which means that MAINVIEW Alarm Manager installed on one system keeps track of your entire sysplex. You can then display a single view that shows exceptions for all MAINVIEW performance monitors within your OS/390 or z/OS enterprise.

**MAINVIEW Alternate Access**

A product that enables MAINVIEW products to be used without TSO by providing access through EXCP and VTAM interfaces.

**MAINVIEW AutoOPERATOR**

A product that uses tools, techniques, and facilities to automate routine operator tasks and provide online performance monitoring, and that achieves high availability through error minimization, improved productivity, and problem prediction and prevention.

---

**MAINVIEW Explorer** A product that provides access to MAINVIEW products from a Web browser running under Windows. MAINVIEW Explorer replaces MAINVIEW Desktop.

### **MAINVIEW Selection Menu**

The ISPF selection panel that provides access to all MAINVIEW window-mode and full-screen mode products.

**MAINVIEW SRM** *See* MAINVIEW Storage Resource Manager (SRM).

### **MAINVIEW SRM Allocation**

A component of MAINVIEW SRM that

- provides control over data set allocation and enforcement of allocation and naming standards
- operates at the system level to intercept abend conditions or standards violations, thus providing services without any JCL changes
- provides tools that aid in the conversion to DFSMS and enhance the DFSMS environment after conversion

### **MAINVIEW SRM Automation**

A component of MAINVIEW SRM that delivers powerful event generation and storage automation technology across the storage enterprise. Used in conjunction with MAINVIEW AutoOPERATOR, automated solutions to perform pool, volume, application, or data set-level manipulation can be created and used in response to any condition or invoked to perform ad hoc requests.

### **MAINVIEW SRM DMS2HSM**

A product that facilitates the conversion of CA-Disk, formerly known as DMS, to HSM.

### **MAINVIEW SRM EasyHSM**

*See* MAINVIEW SRM Allocation.

### **MAINVIEW SRM EasyPOOL**

*See* MAINVIEW SRM Allocation.

### **MAINVIEW SRM EasySMS**

*See* MAINVIEW SRM Allocation.

---

## **MAINVIEW SRM Enterprise Storage Automation**

*See MAINVIEW SRM Automation.*

## **MAINVIEW SRM Reporting**

A component of MAINVIEW SRM that monitors and reports on DASD consumption and allows you to dynamically control DASD utilization. Views enable the DASD administrator to review historic DASD usage and control current and future DASD usage. Physical views of storage devices can be supplemented with user-defined applications that allow for budgeting and measurement by logical groups. MAINVIEW SRM Reporting also provides online monitoring and reporting to help storage managers use DFHSM efficiently

## **MAINVIEW SRM SG-Auto**

Product that provides early warning notification of storage anomalies and automated responses to those anomalies based on conditions in the storage subsystem.

## **MAINVIEW SRM SG-Control**

*See MAINVIEW SRM Reporting.*

## **MAINVIEW SRM StopX37/II**

*See MAINVIEW SRM Allocation.*

## **MAINVIEW SRM StorageGUARD**

*See MAINVIEW SRM Reporting.*

## **MAINVIEW Storage Resource Manager (SRM)**

A suite of products that assist in all phases of OS/390 or z/OS storage management. MAINVIEW SRM consists of products that perform automation, reporting, trend analysis, and error correction for storage management.

**management class** A collection of management attributes, defined by the storage administrator, used to control, among others, the

- release of allocated but unused space
- retention, migration, and backup of data sets
- retention and back up of aggregate groups
- retention, backup, and class transition of objects

**masking** *See* name masking.

**MCDS** *See* Migration Control Data Set (MCDS).

## Migration Control Data Set (MCDS)

A data set that contains information about data sets migrated by HSM.

<b>mount type</b>	The term used to indicate whether a tape or disk is public, storage, or private so that the system knows how the volume is to be used and what type of data is to be stored on it.
-------------------	--

**MLA** *See* Multi-Level Automation (MLA).

### Multi-Level Automation (MLA)

In MAINVIEW SRM Automation, the user-defined, multiple step process that implements automation solutions in a tiered approach, where solutions are invoked one after another until the condition is resolved.

**MVALARM**      *See MAINVIEW Alarm Manager.*

**MVI** MAINVIEW infrastructure, which is the basic architecture that distributes work between workstations and multiple OS/390 targets for BMC Software MAINVIEW products.

<b>name masking</b>	A facility that allows resource selection parameters to select large or small groups of resources using character symbols.
---------------------	--

**OAM** *See* Object Access Method (OAM).

## Object Access Method (OAM)

In the IBM ImagePlus system, a program that provides object storage, object retrieval, and object storage hierarchy management. The Object Access Method isolates applications from storage devices, storage management, and storage device hierarchy management.

<b>online help</b>	The Help information that is accessible online.
--------------------	---

## OS/390 and z/OS Installer

The BMC Software common installation system for mainframe products.

## OS/390 product address space (PAS)

---

The address space containing OS/390 or z/OS data collectors, including the CMF MONITOR Extractor. Used by MAINVIEW for OS/390, MAINVIEW for UNIX System Services, CMF MONITOR, and MAINVIEW SRM products.

**parameter library** Data set consisting of members that contain parameters for specific MAINVIEW products or a support component. There can be several versions:

- the distributed parameter library, called BBPARM
- a site-specific parameter library or libraries

These can be

- a library created by AutoCustomization, called UBBPARM
- a library created manually, with a unique name

**parmlib** *See* parameter library.

### **Partitioned Data Set (PDS)**

A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

### **Partitioned Data Set Extended (PDSE)**

A data set that consists of a directory and zero or more members, just like a PDS. It can be created with JCL, TSO/E, and ISPF, just like a PDS, and can be processed with the same access methods. The directory can expand automatically as needed, up to the addressing limit of 524,286 members. It also has an index, which provides a fast search for member names. Space from deleted or moved members is automatically reused for new members, so you do not have to compress a PDSE to remove wasted space. Each member of a PDSE can have up to 15,728,639 records.

**PAS** *See* product address space (PAS).

**patch** A fix applied to a program to correct an error in program logic or processing. *See also* PTF and PUT.

**PDS** *See* Partitioned Data Set (PDS).

**PDSE** *See* Partitioned Data Set Extended (PDSE).

### **Periodic Update Tape (PUT)**

A tape containing software updates to IBM programs. *See also* PTF.

---

<b>pool</b>	A predefined set of DASD volumes used to store groups of logically related data for more efficient use of DASD. In the context of MAINVIEW SRM, pool is used to mean an SMS Storage Group, a MAINVIEW SRM-defined pool, or a MAINVIEW SRM-defined SMS Pool. <i>See also</i> pooling.
<b>pooling</b>	A technique of managing DASD. Also known as volume or storage pooling. Controlling data set placement by directing data to specified disk volumes based on user criteria. A method used to help the data center manage disk space in a more efficient manner. A term used to indicate that certain groups of disk devices are being dedicated to certain types of data or for specific users. Some of the more common pools created are for test data, production data, database files, system data sets, and work (temporary) packs.
<b>pop-up window</b>	A window containing Help information that, when active, overlays part of the window area. A pop-up window is displayed when you issue the HELP command while working in windows-mode.
<b>primary command field</b>	The field in the control area where commands can be entered. In MAINVIEW, the command line is considered the primary command field.
<b>primary space allocation</b>	The amount of space requested by a user for a data set when it is created. <i>Contrast with</i> secondary space allocation.
<b>procedure library</b>	<p>A data set consisting of members that contain executable procedures used by MAINVIEW products. These procedures are execute command lists (EXECs) that automate site functions. There can be several versions:</p> <ul style="list-style-type: none"> <li>• the distributed parameter library, called BBPROC</li> <li>• a site-specific parameter library or libraries</li> </ul> <p>These can be</p> <ul style="list-style-type: none"> <li>• a library created by AutoCustomization, called UBBPROC</li> <li>• a library created manually, with a unique name</li> </ul> <p>The site-created EXECs can be either user-written or customized EXECs from BBPROC.</p>
<b>product address space (PAS)</b>	In MAINVIEW products, the address space that contains data collectors and other product functions. <i>See also</i> OS/390 product address space (PAS) and BBI subsystem product address space (BBI-SS PAS).

---

**profile library**

A data set consisting of members that contain profile information and cycle refresh definitions for a terminal session connected to a BBI-SS PAS. Other members are dynamically created by MAINVIEW applications. There can be several versions:

- the distributed profile library, called BBPROF
- a site-specific profile library or libraries

These can be

- a library created by AutoCustomization, called SBBPROF
- a library created manually, with a unique name

The site library is a common profile shared by all site users. The terminal session CLIST creates a user profile automatically if one does not exist; it is called userID. BBPROF, where userID is your logon ID. User profile libraries allow each user to specify unique PF keys, CYCLE commands, target system defaults, a Primary Option Menu, and a unique set of application profiles.

**PTF**

*See* Program Temporary Fix (PTF). A patch (fix) to a program.

**private volume**

The mount attribute for disk packs normally reserved for relatively permanent data storage. A mounted volume that the system can allocate to only if it is requested by its specific volume serial number. *See also* public volume and storage volume.

**Program Temporary Fix (PTF)**

A patch (fix) to a program. PTFs are distributed on PUTs. An APAR contains the description of a problem. One or more PTFs might be issued as corrective action. For example, one PTF is generally issued for each release level that is in full support. However, only one BAG (APAR) is created. *See also* APAR, PUT, and patch.

**public volume**

The mount attribute for disk packs reserved for temporary data. Temporary data usually exists only for the life of a job and is deleted when the job ends. The operating system might automatically select any public volume to allocate temporary data. *See also* private volume and storage volume.

**PUT**

*See* Periodic Update Tape (PUT).

**QA**

*See* Quality Assurance (QA).

**Quality Assurance (QA)**

The process of testing programs and documentation to determine that they function as specified in the design.

---

**query** In MAINVIEW, one of two constituent parts of a view; the other is form. A query defines the data for a view; a form defines the display format. *See also* form and view.

**queue** A group of items or jobs waiting to be acted upon by the computer are held in a queue or buffer storage area. The arrangement of items or jobs determines their processing priority. Sometimes priority codes are assigned numerically so that critical jobs are processed ahead of less important ones.

**RACF** *See* Resource Access Control Facility (RACF).

**RAID** *See* Redundant Arrays of Independent Disks (RAID).

**RAMAC** A type of disk storage system from IBM. *See also* RVA.

### **RAMAC Virtual Array (RVA)**

An online, random access disk array storage system composed of disk storage and control unit combined into a single frame.

**real-time data** The data as it exists at the moment of inquiry. Real-time data is recorded during the smallest unit of time for data collection. *Contrast with* historical data and interval data. *See also* current data.

### **Record Level Sharing (RLS)**

The sharing of records between address spaces. Sharing can take place on the same machine or on multiple machines.

### **Redundant Arrays of Independent Disks (RAID)**

A generic term for a disk drive made up of a large number of small platters and heads. In networking and mission-critical applications, a method of using several hard disk drives in an array to provide fault tolerance in the event that one or more drives fail.

### **Resource Access Control Facility (RACF)**

An IBM licensed data security software package that controls access to a computer system's resources and limits the actions that a user can take. RACF authorizes access to resources, logs unauthorized access attempts, and logs accesses to protected data sets.

**REXX** *See* Restructured Extended Executor (REXX).

### **Resource Measurement Facility (RMF)**



---

An IBM licensed online performance and resource monitor software for MVS systems. RMF monitors selected areas of system activity and presents the data collected in the form of printed reports, SMF records, and/or display reports.

### **Restructured Extended Executor (REXX)**

In IBM, a general-purpose, procedural language for end-user personal programming, designed for ease by both casual general users and computer professionals. It is also useful for application macros. It can replace TSO CLISTS or be used in conjunction with them.

**RLS**                      *See* Record Level Sharing (RLS).

**RLST**                    *See* rule list.

**RMF**                      *See* Resource Measurement Facility (RMF).

**row**                      In MAINVIEW SRM, a horizontal component of a view or display comprising all the fields pertaining to a single device, address space, user, and so on.

**rule list**                A list of parameters that select resources that are modified with specific values in action parameters. A rule list is invoked for the group of resources already selected by the filter list; individual rules in the list can then select all or some resources from that group for processing. The function definition identifies the rule list to use. *See also* filter list.

**RVA**                      *See* RAMAC Virtual Array (RVA).

**SAS**                      *See* Statistical Analysis System (SAS).

### **secondary space allocation**

The amount of additional space requested by the user for a data set when primary space is full. *Contrast with* primary space allocation.

### **selection parameters**

In MAINVIEW SRM, parameters that define resource selection using filter list and rule list members. Selection parameters are defined in INC (include) and EXC (exclude) statements. Selection parameter values are compared to OS/390 properties; if the selection parameter is equal to the OS/390 value, the resource is selected for processing by the MAINVIEW SRM function. *Contrast with* SET keywords.

---

<b>SET keywords</b>	The parameters that are specified on SET statements that begin with the keyword SET and define values that either control MAINVIEW SRM operation or set a value that is subsequently used by OS/390 or DFSMS. <i>See also</i> SET statements. <i>Contrast with</i> selection parameters.
<b>SET result group</b>	A group of records matching the same AUTO function FLST/RLST SET statement. The AUTO function FLST/RLST processes slightly differently than most other MAINVIEW SRM functions. The AUTO functions FLST/RLST create a group of records according to the INC/EXC statements of the SET. The SET statement keywords then determine how to process the group of records called the SET result group. This allows sorting of the result group and actions implemented in sort order. <i>See also</i> SET statements.
<b>SET statement</b>	A statement that begins with the keyword SET and defines values that either control MAINVIEW SRM operation or set a value that is subsequently used by OS/390 or DFSMS. <i>See also</i> action parameters.
<b>SG-Auto</b>	<i>See</i> MAINVIEW SRM SG-Auto.
<b>SG-Control</b>	<i>See</i> MAINVIEW SRM Reporting.
<b>shared parmlib environment</b>	A common parameter library that is used by multiple systems in a sysplex.
<b>Shark</b>	<i>See</i> Enterprise Storage Server (ESS).
<b>Skeleton Tailoring Facility</b>	A facility in MAINVIEW AutoOPERATOR that allows skeleton JCL to be used during job submission. Skeleton JCL can contain variables within the JCL statements to be substituted with data values at job submission time. Directive statements can be used in the skeleton JCL to cause the repetition of a set of skeleton statements. This facility functions similar to the ISPF skeleton tailoring facility.
<b>SMF</b>	<i>See</i> System Management Facilities (SMF).
<b>SMP</b>	<i>See</i> System Modification Program (SMP).
<b>SMP/E</b>	<i>See</i> System Modification Program Extended (SMP/E).
<b>SMS</b>	<i>See</i> Data Facility System Management Services (DFSMS).
<b>SMS class</b>	A list of attributes SMS applies to data sets having similar allocation (Data Class), performance (Storage Class), or availability (Management Class) needs.

---

**SMS-managed data set**

A data set that has been assigned a Storage Class.

**SMS-managed volume**

An online volume that is defined in the active SMS configuration. An SMS-managed volume can be an SMS volume or a non-SMS volume.

**SMS volume**

A volume that contains only SMS data sets (a fully converted SMS volume) or a partially converted SMS volume.

**solution**

In the context of MAINVIEW SRM automation, a solution is an action that is taken to affect a condition, hopefully resolving it. Solutions are defined in AUTO function FLST/RLST SET statements and determine the automated action to take and the resources to take it on. Solutions are requested through the AUTO console commands. A solution either submits a job to manipulate resources or generates events that can then further analyze the condition.

**space administrator**

An individual that is assigned to perform DASD storage administration functions.

**started task**

An address space that is running one or more programs (jobs) that were initiated manually using the START operator command.

**Statistical Analysis System (SAS)**

An SAS Institute-licensed software package used for analyzing data.

**stem variables**

A REXX facility, supported in MAINVIEW AutoOPERATOR REXX EXECs and the Skeleton Tailoring Facility, where variable names end with a period followed by a number, such as &GROUP.1. This configuration allows each variable to actually represent a table or array of data, with the zero variable containing the number of entries in the array. For example, &GROUP.0 = 5 would indicate variables &GROUP.1 through &GROUP.5 exist.

**StopX37/II**

*See* MAINVIEW SRM Allocation.

**storage administrator**

A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

**storage class**

A list of DASD storage performance, security, and availability service level requirements for an SMS-managed data set.

---

<b>storage group</b>	A collection of storage volumes and attributes defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD or tape volumes treated as a single object storage hierarchy.
<b>StorageGUARD</b>	<i>See</i> MAINVIEW SRM Reporting.
<b>storage management</b>	The task of managing auxiliary storage resources (tape and DASD) for an installation.
<b>storage pool</b>	A predefined set of DASD volumes most often used for storage of short term or temporary data. <i>See also</i> pool.
<b>storage subsystem</b>	<p>A widely-used term that can mean, among other things:</p> <ul style="list-style-type: none"> <li>• a storage system (the whole box)</li> <li>• the portion of a processor complex that provides data storage and handles input and output for the storage media</li> <li>• a subsystem as defined to HCD containing up to 64 volumes</li> </ul>
<b>storage volume</b>	A mount attribute for disk packs reserved for intermediate results, usually for data that is not going to be retained for long term use. One of a group of mounted volumes the system can allocate for use when a job specifies an esoteric or generic unit name or when a job does not request a specific volume by its serial number. <i>See also</i> public volume and private volume
<b>subsystem</b>	A system that is subordinate to the main system.
<b>summary view</b>	A view created from a tabular view using the Summarize option in view customization. A summary view compresses several rows of data into a single row based on the summarize criteria.
<b>SVOS</b>	<p>A started task that runs all MAINVIEW SRM products. SVOS is the PAS that manages the connection and communication with the CAS. The SVOS PAS provides</p> <ul style="list-style-type: none"> <li>• connection for real-time monitor, administration, and common views</li> <li>• dynamic activation and deactivation of MAINVIEW SRM components</li> <li>• communication with MAINVIEW SRM components</li> </ul> <p><i>See also</i> started task, CAS, and PAS.</p>
<b>sysplex</b>	A sysplex (systems complex) that consists of multiple systems coupled together by hardware elements and system services.

---

## System Management Facilities (SMF)

A component of OS/390 that collects input/output (I/O) statistics provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

## System Modification Program (SMP)

A type software used to make modifications to the IBM operating system programs. SMP contains tracking features that keep records of all system modifications. SMP provides an audit trail of system alterations.

## System Modification Program Extended (SMP/E)

An enhanced version of SMP. *See also* System Modification Program (SMP).

**system parameters** *See* global parameters.

## terminal session (TS)

A single point of control for MAINVIEW products, allowing data manipulation and data display and providing other terminal user services for MAINVIEW products. The terminal session runs in a user address space (either a TSO address space or a stand-alone address space for EXCP/VTAM access).

## Time Sharing Option (TSO)

A software component supported by the OS/390 operating system that supplies concurrent time sharing among many users from remote terminals.

**threshold** A specified value that is used to determine whether the data in a field meets specific criteria.

**Top Secret** *See* CA-Top Secret.

**TS** *See* terminal session.

**TSO** *See* Time Sharing Option (TSO).

**UAS** *See* user address space (UAS).

**UBBPARM** *See* parameter library.

**UBBPROC** *See* procedure library.

**UBBSAMP** *See* sample library.

---

**unload** To download data sets from the tape or FTP site to DASD.

**user address space (UAS)**

In MAINVIEW, the address space that runs the terminal session (TS) in TSO, VTAM, or EXCP mode.

**view** The formatted data within a MAINVIEW window, acquired from a product as a result of a view command or action. A view consists of two parts: query and form. *See also* query and form.

**view help** The online Help that describes the purpose of a view. To display view Help, place the cursor on the view name on the window information line and press PF1 (HELP).

**Virtual Input/Output (VIO)**

A type of storage group that allocates data sets to paging storage, which simulates a DASD volume. VIO storage groups do not contain any actual DASD volumes. *See also* storage group.

**Virtual Storage Access Method (VSAM)**

An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

**Virtual Tape Control System (VTCS)**

The StorageTek product for virtual tape.

**VIO** *See* Virtual Input/Output (VIO).

**Volume Table of Contents (VTOC)**

The area of a disk used to store the directory (index) of all non-VSAM data sets or files that are contained on that disk volume.

**volume switch** A process during data set creation or extension in which the original disk volume becomes full or a data set has reached 16 extents on that volume and another volume is selected to continue building the data set. The data set then exists in pieces or extents on more than one disk pack. Pointers exist at the end of each extent to indicate where the next piece of the data set is located.

**VSAM** *See* Virtual Storage Access Method (VSAM).

---

<b>VTCS</b>	<i>See</i> Virtual Tape Control System (VTCS).
<b>VTOC</b>	<i>See</i> Volume Table of Contents (VTOC).





---

# Index

## A

AA\_AMODE E-1, E-18  
AA\_APPL E-1, E-18  
AA\_ASTAT E-1  
AA\_CDATE E-1, E-19  
AA\_HSMC E-1, E-19  
AA\_HSMH E-1, E-19  
AA\_KHSM E-1, E-20  
AA\_KTEMP E-1, E-20  
AA\_KVSAM E-1, E-20  
AA\_LDATE E-1, E-20  
AA\_PERMC E-2, E-20  
AA\_PERMH E-2, E-21  
AA\_PERMM E-2, E-21  
AA\_PERMP E-2, E-21  
AA\_PHSM E-2, E-21  
AA\_PTEMP E-2, E-21  
AA\_PVSAM E-2, E-22  
AA\_TEMPC E-2, E-22  
AA\_TEMPH E-2, E-22  
AA\_TEMPM E-2, E-22  
AA\_TEMPP E-2, E-23  
AA\_UFLDn E-2, E-23  
AA\_UNAME E-2, E-23  
AA\_VLCNT E-2, E-23  
AA\_VSAMC E-2, E-24  
AA\_VSAMH E-2, E-24  
AA\_VSMM E-2, E-24  
AA\_VSAMP E-2, E-24  
AA\_WTHRS E-2, E-25  
ABEND 9-4  
AC\_CODE E-3, E-25

ACF2 E-91, E-140  
ACF2USER E-3, E-25  
ACT\_COUNT E-3, E-26  
ACT\_EVENT E-26  
ACT\_EVENTID E-3  
ACT\_JOB E-3, E-27  
ACT\_SUM\_FLD E-3, E-28  
ACT\_SUM\_LIMIT E-3, E-28  
action parameters A-2  
    defined A-2  
    relationships A-4  
activate  
    dynamic 2-2  
activating system software into MVS 2-2  
ACTIVE 4-3, D-108, D-109  
AD\_ALVL1 E-3, E-28  
AD\_ALVL2 E-3, E-29  
AD\_ALVL3 E-3, E-29  
AD\_ALVL4 E-3, E-29  
AD\_BLKEF E-3, E-29  
AD\_BLKESZ E-3, E-29  
AD\_BLKTR E-3, E-30  
AD\_CASPL E-3, E-30  
AD\_CAT E-3, E-30  
AD\_CDATE E-3, E-30  
AD\_CHG E-3, E-31  
AD\_CISPL E-3, E-31  
AD\_DAYS E-3, E-31  
AD\_DCLAS E-3, E-31  
AD\_DOCC E-32  
AD\_DOP E-3  
AD\_DSN E-3, E-32  
AD\_DSORG E-3, E-32  
AD\_EXTS E-3, E-32

---

AD_GROUP E-3, E-33	AP_LIFUL E-5, E-44
AD_LDATE E-4, E-33	AP_LPRIC E-5, E-44
AD_LRECL E-4, E-33	AP_LPRIT E-5, E-44
AD_MCLAS E-4, E-33	AP_LREEC E-5, E-44
AD_POOL E-4, E-34	AP_LREED E-5, E-45
AD_POOLT E-4, E-34	AP_LREET E-5, E-45
AD_PTyp E-4	AP_LREEV E-5, E-45
AD_PUSED E-4, E-34	AP_LREEX E-5, E-45
AD_REBLK E-4, E-34	AP_LVFRG E-5, E-46
AD_RECFCM E-4, E-35	AP_LVFUL E-5, E-46
AD_SCLAS E-4, E-35	AP_PERFL E-5, E-46
AD_SIZE E-4, E-36	AP_POOL E-5, E-46
AD_SMSI E-4, E-36	AP_TSIZE E-6, E-46
AD_TRKSA E-4, E-36	AP_TYPE E-6, E-47
AD_TRKSF E-4, E-37	AP_USIZE E-6, E-47
AD_TRKSU E-4, E-37	AP_VOLC E-6, E-47
AD_VOL E-4, E-37	AP_VOLD E-6, E-47
AD_VOLSQ E-4, E-37	asterisk
AD_XDATE E-4, E-38	comment line A-3
ADR 6-3, D-99, D-100	AUTO_MXTSK D-23
AIX	AUTOJ_OINDX D-3, D-23
in VSAMDEF E-145	AUTOLEV E-6, E-48
ALCTYPE E-4, E-38	Automation 2-9
Allocation 2-9	AUTO-MXTSK D-3, D-23
ALLTAPE D-80	AUTOPROC D-3, D-24
ALTPOOL E-4, E-38	AV_CTIGC E-6, E-48
AOO_SUBSYS D-3	AV_CTIGT E-6, E-48
AP_CTIGC E-4, E-39	AV_DEV E-6, E-48
AP_CTIGT E-4, E-39	AV_FRAGI E-6, E-49
AP_FREEC E-4, E-39	AV_FREEC E-6, E-49
AP_FREED E-4, E-39	AV_FREED E-6, E-49
AP_FREET E-4, E-40	AV_FREET E-6, E-49
AP_FREEV E-4, E-40	AV_FREEV E-6, E-50
AP_FREEX E-4, E-40	AV_FREEX E-6, E-50
AP_FSIZE E-4, E-40	AV_FSIZE E-6, E-50
AP_GROUP E-5, E-41	AV_FULL E-6, E-50
AP_HDFUL E-5, E-41	AV_IPCTF E-6, E-51
AP_HFULL E-5, E-41	AV_ISIZE E-6, E-51
AP_HIFUL E-5, E-41	AV_LPRIC E-6, E-51
AP_HREEC E-5, E-42	AV_LPRIT E-6, E-51
AP_HREED E-5, E-42	AV_MNT E-6, E-52
AP_HREET E-5, E-42	AV_PCNT E-6, E-52
AP_HREEV E-5, E-42	AV_PID E-7, E-52
AP_HREEX E-5, E-42	AV_POOLn E-7, E-52
AP_HVFRG E-5, E-43	AV_PTypn E-7, E-52
AP_HVFUL E-5, E-43	AV_MSGP E-7, E-53
AP_LDFUL E-5, E-43	AV_SMSI E-7, E-53
AP_LFULL E-5, E-43	AV_TSIZE E-7, E-53

---

AV\_UCB E-7  
AV\_USIZE E-7, E-54  
AV\_VOL E-7, E-54  
AV\_VPCTF E-7  
AV\_VPTCF E-54  
AV\_VTOCF E-7, E-55  
AV\_VTOCI E-7, E-55  
AV\_VTOCZ E-7, E-55  
AV\_VVDSF E-54  
AVL E-7, E-56

## B

BACKCMD E-7, E-56  
BACKUP E-7, E-56  
BBI3\_SSID D-3, D-25  
BBPARM 3-3  
BBSAMP members  
    SGPPROC D-48  
BCDSn D-3, D-25  
BESTFIT E-143  
BLKINPUT D-3  
BLKOLDSR D-3, D-26  
BLKSIZE E-7, E-57  
BUFSP E-7, E-57

## C

CA E-140  
CA-ACF2 E-91  
cached devices D-29, D-69  
CAL D-3, E-7, E-57  
CALAGE E-7, E-58  
calendar  
    parameter D-26  
    SMCAL parameter D-104  
CANCEL 2-7  
CANDIDATE E-7, E-58  
CAT E-7, E-58  
CATALOG E-7, E-58  
catalog name D-82, E-7  
CA-Top Secret E-91, E-140  
CHECK D-3, D-26  
CISIZE E-7, E-59  
CLUSTER  
    in VSAMDEF E-145

COLT E-121  
common command structure 2-14  
communicating with MAINVIEWE SRM  
    components 2-14  
COMP E-7, E-59  
comparison operators A-6  
CONFIG\_MXTSK D-4, D-27  
console  
    commands  
        DISPLAY 2-21  
        REFRESH 2-22  
        SET 2-23  
        VSCAN 2-24  
console commands  
    DISPLAY 2-21  
    REFRESH 2-21  
    SET 2-21  
    SVOS CONFIG,DIAG 2-21  
    SVOS CONFIG,REFRESH 2-21  
CONTIG E-7, E-59  
controlling MAINVIEW SRM components 2-14  
controlling secondary reduction D-41  
CRITBIAS E-7, E-60  
CRITDSN E-143  
CRITEMC E-7, E-60  
CRITFAIL E-7, E-60  
CRITLIST D-4, D-27, E-7, E-61  
CURDAY E-7, E-61  
CURSPACE E-7, E-61  
CURTIME E-7, E-61

## D

DADSM\_FUNC E-8, E-62  
DADSMEX D-4, D-27  
DASDGENR D-4, D-28  
DASDPOOL D-102  
data sets  
    CURSPACE parameter E-61  
    size 3-5  
DATACLAS 3-6, E-8, E-62  
DATEFMT D-4, D-28  
DB2 E-67  
DCTYPE D-4, D-29  
DD 3-7, E-8, E-63  
DD statement  
    PROIGN

---

- override 9-3
- suppress jobstep activity D-38
- DD statements
  - PROSD\$M\$ 9-10
  - PRODUMP 9-10
  - PROIGN 9-10, D-38
  - PROMTRCE 9-10
  - PROTRACE 9-10
- DEBUG 9-4
- Defining 4-1
- defining pools D-99, D-103
- definition member
  - parameters 3-5
  - SMCAL\$xx 8-2
- definition members
  - SMFUNCxx 4-2
  - SMVARSxx 7-2
- DEFUNIT E-8, E-63
- DESC 4-3, D-108, D-111
- DEVTYPE E-8, E-63
- DFREORGPRC D-4, D-29
- DIAG D-4, D-29
  - IGNORE parameter 9-3
- DIAGMSDD 9-10, D-4, D-30
- diagnostic parameters 9-2, D-112
- DIR E-8, E-63
- DISPLAY 2-21, D-4, D-30
- DISPn E-8, E-64
- DMYUNIT D-4, D-30
- DP\_RENAME D-4, D-31
- DPO E-143
- DPORDEF E-8, E-64
- DPORMAX E-8, E-65
- DPORMIN E-8, E-65
- DPORSEP E-8, E-66
- DPOWIND E-8, E-66
- DSN E-8, E-66
- DSNAME E-8, E-67
- DSNn E-8, E-67
- DSNTYPE E-8, E-67
- DSORG E-8, E-68
- DSTYPE E-8, E-68
- dualcopy D-29
- DUMP 9-4
- DUMP command 2-16
- DUMPPDD 9-10, D-5, D-31
- DYNALLOC E-8, E-68
- dynamic activate 2-2

- overview 2-2

## E

- EasyHSM renamed HSM collector 1-2
- Enterprise Storage Automation renamed
  - MANIVIEW SRM Automation 1-2
- ENVIR E-8, E-69
- ERASE E-8, E-69
- ETS\_ID D-31
- EVENTID 5-2, D-113, E-70
- EVNT D-5, D-32
- EXPDT 3-6, E-8, E-70
- EXTENT E-8, E-70
- EZ menus 1-4
- EZcmd menu 1-5

## F

- fastwrite D-29
- FDRIAM D-5, D-32
- field-to-field compares A-10
- FILESEQ E-8, E-71
- filter list and rule list parameters
  - reference E-1
- filter list parameters 5-2
  - examples 5-9
  - reference E-1
- filter lists
  - defined 3-8
- FLST 4-2, D-108, D-109
- FORCE E-8, E-71
  - allow from console 2-6
  - command 2-17
  - parm on SVOS 2-6
- FORCECAT 5-7
- FORPLEXNAME 3-11, D-32
- FORSMFID 3-11, D-33
- FORSYSID 3-11, D-33
- FRAGCNTL 5-7
- FREE 8-2, D-105
- FUNC D-5
- FUNCTION 9-4, D-112, E-8, E-71
- functions
  - define and activate 4-1
  - definition examples 4-3

---

## G

GDGVER E-8, E-72  
general syntax rules A-1  
global parameters  
    reference D-1  
global parameters - SMMSYSxx D-2

## H

HDPODAYS E-8, E-74  
HDPOETIM E-8, E-74  
HDPORDEF E-8, E-72  
HDPORMAX E-8, E-72  
HDPORMIN E-9, E-73  
HDPORSEP E-9, E-73  
HDPOSTIM E-9, E-74  
hierarchal structure of PDS members 3-2  
HISTDAYS D-5, D-34  
HISTDPO E-143  
HLOGAUTH D-5, D-34  
HLOGAUTM D-5, D-35  
HLOGCOLL D-5, D-35  
HLOGINDX D-5, D-35, D-36, D-37  
HLOGLIM D-5, D-36  
HLOGPRIM D-5  
HLOGTASK D-5, D-37  
HLOGUNIT D-5  
HLOGYDSN D-5, D-37  
HLQ E-9, E-75  
HSM E-9, E-75  
HSMACTID D-5, D-38  
HSMDELET 5-7  
HSMDSN E-9, E-75

## I

IAM E-67  
IGNORE 9-4  
    parameter 9-3  
IGNOREDD 9-10, D-6, D-38  
IMBED E-9, E-75

## J

JCLEXT D-6, D-39  
JCLUREQ D-6, D-39  
JOB 9-4, D-112, E-9, E-76  
JOBACCT 3-7  
JOBACCTn E-9, E-76  
JOBCLASS E-9, E-76  
JOBSDAY E-9, E-76  
JOBSTIME E-9, E-77  
JOBTYPE E-9, E-77

## L

LABELTYP E-9, E-77  
LEVEL E-9, E-78  
LIMIT 3-9, E-9, E-78  
limiting secondary reduction D-41  
LLQ E-9, E-78  
LRECL E-9, E-78

## M

MAINVIEW SRM Allocation 2-14  
MAINVIEW SRM Automation 2-14  
MAINVIEW SRM component identifier  
    SVALLOC 2-14  
    SVESA 2-14  
    SVHSM 2-14  
    SVSGA 2-14  
    SVSGD 2-14  
    SVSGP 2-14  
MAINVIEW SRM Parameters 3-6  
MAINVIEW SRM Reporting 2-14  
masking syntax A-6  
master system definition member  
    SMMSYSxx 3-3  
master system member parameters D-2  
maximum volumes E-136  
    SPACVOLA parameter E-16  
MAXQLF E-9, E-79  
MAXSIZE 3-7, E-9, E-79  
MAXSPACE E-143  
MAXVOL D-6, D-40  
MCDSn D-6  
menu

---

EZ 1-4  
EZcmd 1-5  
EZSRM 1-4  
using 1-4  
messages  
    written to the SMF data set B-2  
MGMTCLAS E-9, E-79  
MIGCMD E-9, E-80  
MIGDAYS E-9, E-80  
MIGRATE E-9, E-80  
MINQLF E-9, E-81  
ML2 E-9, E-81  
MNTYPE E-9, E-81  
MODDELET 5-7  
MODE 5-2, 5-4, E-9, E-82  
    parameter 5-4  
MODIFY  
    commands 2-15  
MODTRC 9-4  
MODTRCDD 9-10, D-6, D-41  
MODULE 9-4, D-112  
MON-SUN 8-2, D-105, D-106  
mount type  
    parameter E-81  
MREDUCE D-6, D-41  
MSG 3-6, 3-9, 4-3, 5-3, 5-5, D-108, D-110, E-9,  
    E-83  
MSGID D-6, D-42  
MSGLVL D-6, D-42  
MSGPREF D-6, D-42  
multiple subsystems  
    support 2-14  
MVI infrastructure 3-10

## N

NAME 4-2, D-108  
name masking 3-4, 3-9, A-6  
navigation 1-4  
NEWACCT E-9, E-83  
NEWAPPL E-83  
NOCATDYN D-6, D-43  
NOCATLG2 E-9  
NOCATPFX D-6, D-43  
NOCATPRG D-7, D-43  
NOCATSEC D-7, D-44  
NOCATSMS D-7, D-44

NOCATTIM D-7  
NOCATVOL D-7, D-45  
NOCATWHEN D-7, D-46, E-9, E-84  
NOCHECK E-9, E-85  
NOVERIFY  
    parm on SVOS 2-6  
NQVAL E-10, E-85  
NUNIT E-10, E-85  
NVOL E-10, E-86  
NVOLINDX E-10, E-86  
NVOLMAX E-10, E-86

## O

OCDS D-7, D-46  
OLDACCT E-10, E-86  
OLDDSN E-10, E-87  
OLDHLQ E-10, E-87  
OPENEMPT 5-7  
OPER E-10, E-87  
OPMHLQ D-7, D-46  
ORIGDATA D-8, D-47  
ORIGUNIT E-10, E-87  
ORIGVOL E-10, E-88  
override PROIGN DD statement  
    IGNORE 9-3  
OWNER E-10, E-88

## P

parameter description 2-5  
parameter relationships A-4  
parameters  
    action A-2  
    calendar settings, SMCALS D-104  
    filter and rule list descriptions E-1  
    global parameter SMMSYSxx D-2  
    masking syntax A-6  
    selection A-2  
    values, definition of A-2  
PASSWORD D-8, D-47, D-48  
PCT E-88  
PCTI E-10  
PERCENT E-143  
PERFRM\_PRC D-8  
PGM 9-4, E-10, E-88

---

PGMRNAME E-10, E-88  
POOL D-8, D-48, E-10, E-89  
pool  
    defining parameters with SMPOOL D-99  
POOLNAME 6-3, D-99, D-100, D-103  
    description D-99  
pools  
    examples 6-4  
PQTY E-10, E-89  
PRISPACE E-10, E-90  
PRO\$D\$M\$  
    activates WTO message trace 9-10  
PROCOLD D-8, D-49  
PROCSTEP E-10, E-90  
PRODUMP  
    produces DUMP 9-10  
PROGRAM D-112  
PROIGN  
    deactivates tracing 9-10  
PROMTRCE  
    activates entry/exit tracing 9-10  
PROTRACE  
    activates tracing 9-10  
PURGE E-10, E-90  
PWDDEL E-10, E-90

## Q

QUALL E-10, E-91  
QUALn E-10, E-91

## R

RACF 3-9, E-10, E-91, E-140  
RACFGRP E-10, E-91  
RACFUID E-10, E-91  
RAIDDEVTYPE E-10, E-92  
RECFM E-10, E-92  
REORG E-10, E-93  
REFAGE E-10, E-93  
REFRESH command 2-22  
REFVOL E-10, E-93  
REJECT D-8, D-49, E-11, E-94  
RELEASE E-11, E-94  
REORG E-11, E-94  
REORG\_NSMS E-11, E-94

REORG\_PROC E-11, E-95  
REORG\_SMS E-11, E-95  
REPL E-11, E-95  
REPLACE 3-9, E-11, E-96  
Reporting 2-9  
REQTYPE D-8, D-49  
RETPD E-11, E-96  
REUSE E-11, E-96  
RLS D-8, D-50  
RLSE E-11, E-96  
RLST 4-3, D-108, D-109  
ROUND E-11, E-97  
rule list parameters 5-7  
    examples 5-9  
    reference E-1  
rule lists  
    defined 3-8

## S

SCAN E-11  
SCAT D-8, D-50  
secondary reduction  
    limitations D-41  
SECSpace E-11, E-97  
security E-91  
selection criteria ordering A-11  
selection parameters  
    comparison operators A-6  
    defined A-2  
SEP E-11, E-97  
SET command 2-23  
    description D-2  
SET statement filter list parameters 5-2  
SET statements  
    defined A-2  
setting the calendar parameter  
    SMCALs D-104  
SG\_INITPOOL D-8, D-50  
SG\_INITVOL D-8, D-51  
SG\_IXFPNTVL D-51  
SG\_MAXACCT D-9, D-51  
SG\_MAXPOOL D-9, D-51  
SG\_MAXSSDSZ D-9, D-52  
SG\_READNTVL D-9, D-52  
SG\_RETRYLIM D-9, D-52  
SG\_SIBSTK D-9

---

SG_SPACHLDR D-9, D-53	SGDP_POOL E-12, E-106
SG_SUBTASKS D-9, D-53	SGDP_RSVD E-12, E-106
SG_VVDSINFO D-10, D-54	SGDP_RVAARC E-12, E-107
SG_WRITNTVL D-10, D-54	SGDP_RVAFNC E-12
SGA_ENQSCOP D-10, D-54	SGDP_RVAFSC E-12, E-107
SGACMD D-10, D-54	SGDP_RVAFSNC E-107
SGASCAN D-10, D-55	SGDP_RVAIND E-12, E-108
SGASIM D-10, D-55	SGDP_RVANCL E-12, E-108
SG-Auto 2-14	SGDP_SPACDA E-12
SGC_ADDEXIT D-10	SGDP_SPACDN E-12, E-108
SGC_CHKEXIT D-10, D-55	SGDP_SPACEN E-12, E-109
SGC_DEFEXIT D-10	SGDP_SPACNM E-12, E-109
SGC_FUNC E-11, E-98	SGDP_SPACNS E-12, E-109
SGC_KEYEXIT D-10, D-56	SGDP_SPACQA E-12, E-109
SGC_SECEXIT D-10, D-56	SGDP_SPACQN E-13, E-110
SGC_SELEXIT D-10, D-56	SGDP_TYPE E-13, E-108
SGC_STOGRP D-11, D-57	SGDP_VOL#DA E-13, E-110
SGC_STORCLS D-11, D-57	SGDP_VOL#DN E-13, E-111
SGCDSN D-11, D-57	SGDP_VOL#EN E-13, E-111
SG-Control renamed application collector 1-2	SGDP_VOL#N E-13
SGD_PROCNM D-11, D-57	SGDP_VOL#NM E-111
SGD_SMFID D-11, D-58	SGDP_VOL#NS E-13, E-111
SGDA_ALNV E-11, E-98	SGDP_VOL#QA E-13, E-112
SGDA_ALV E-11, E-99	SGDP_VOL#QN E-13, E-112
SGDA_AVAIL E-11, E-99	SGDPROCCNMn D-58
SGDA_GRP E-11, E-99	SGDPROCNMn D-12
SGDA_IDLE E-11, E-100	SGDSMFIDn D-12
SGDA_NVDS E-11, E-100	SGDV_ALREXT E-112
SGDA_VSD E-11, E-100	SGDV_ALREXT=nnnnn E-13
SGDCOLLECT 6-3, D-11, D-58, D-101	SGDV_FRAGI E-13, E-112
SGDCOLLECTn D-58, D-99	SGDV_FRCYL E-13, E-113
SGDP_ALNV E-11, E-101	SGDV_FREXT E-13, E-113
SGDP_ALV E-11, E-101	SGDV_FRVIR E-13, E-113
SGDP_AVAIL E-11, E-101	SGDV_IDTR E-13, E-113
SGDP_DS#DN E-12, E-102	SGDV_LREXT E-13, E-114
SGDP_DS#EN E-12, E-102	SGDV_LREXTT E-13, E-114
SGDP_DS#NM E-12	SGDV_NDS E-13, E-114
SGDP_DS#NS E-12, E-103	SGDV_NF0DSC E-13, E-114
SGDP_DS#QA E-12, E-103	SGDV_PERFUL E-13, E-115
SGDP_DS#QN E-12, E-104	SGDV_PHYID E-13, E-115
SGDP_FRAGI E-12, E-104	SGDV_PHYIDT E-13, E-116
SGDP_IDLE E-12, E-104	SGDV_POOL E-13, E-116
SGDP_NCLPER E-12, E-104	SGDV_POOL1 E-13, E-116
SGDP_NNV E-12, E-104, E-105	SGDV_PTYP E-13, E-117
SGDP_NV E-12, E-105	SGDV_RSRVDT E-13, E-119
SGDP_NVOL E-12, E-105	SGDV_RVAFDV E-13
SGDP_OFFL E-12	SGDV_RVAFDVI E-117
SGDP_PERFUL E-12, E-106	SGDV_RVAIND E-13, E-117



---

SGDV_RVAPCS E-13, E-117	SGP_MAXCCUS D-13, D-62
SGDV_RVAPCU E-14, E-118	SGP_MAXDIRS D-13, D-62
SGDV_RVASSF E-14, E-118	SGP_MAXDSNS D-13, D-63
SGDV_RVAVOL E-14, E-118	SGP_MAXJOBS D-13, D-63
SGDV_SMALLD E-14, E-119	SGP_MAXLCUS D-13, D-63
SGDV_SSID E-14, E-119	SGP_MAXPOLS D-13, D-63
SGDV_USEXT E-14, E-119	SGP_MAXPTHs D-13, D-64
SGDV_VIRPU E-14, E-120	SGP_MAXPVLS D-13, D-64
SGDV_VIRSZ E-14, E-120	SGP_MAXRRKS D-14, D-64
SGDV_VOL E-14, E-120	SGP_MAXRSFS D-14, D-64
SGDV_VVDSPU E-14, E-120	SGP_MAXSCLS D-14, D-65
SGDV_VVDSSZ E-14, E-121	SGP_MAXVOLS D-14, D-65
SGINITPOOLn D-12	SGP_NCLPERC E-15, E-126
SGINITVOLn D-12	SGP_NRDHIT@ E-15, E-126
SGMAXACCTn D-12	SGP_NRDPSEC E-15, E-127
SGMAXPOOLn D-12	SGP_NWRHIT@ E-15, E-127
SGMAXSSDSZn D-12, D-61	SGP_NWRTPSC E-15, E-127
SGP_@BUSY E-14, E-121	SGP_PENDTIM E-15, E-127
SGP_BESCOLT E-14	SGP_RDFCOMP D-14, D-65
SGP_BESFREE E-14	SGP_RDHIT@ E-15, E-127
SGP_BESTOTL E-14, E-121	SGP_RDSPRSC E-15, E-127
SGP_BESUNCL E-14, E-122	SGP_READ@ E-15, E-128
SGP_CFWHIT@ E-14, E-122	SGP_RESERV@ E-15, E-128
SGP_CFWPRSC E-14, E-122	SGP_RESPTIM E-15, E-128
SGP_CHPID E-14, E-122	SGP_RSFNAME E-15
SGP_CNTLUID E-14, E-122	SGP_SIBSTK D-14, D-66
SGP_CONNTIM E-14, E-123	SGP_SMF42 D-14
SGP_CUBSYDL E-14, E-123	SGP_SRDHIT@ E-15, E-128
SGP_DFWHIT@ E-14, E-123	SGP_SRDPRSC E-15, E-129
SGP_DFWPRSC E-14, E-123	SGP_SWRHIT@ E-15, E-129
SGP_DISCTIM E-14, E-123	SGP_SWRPRSC E-15, E-129
SGP_DP@BUSY E-14, E-123	SGP_TRACE D-14, D-66
SGP_DPBSYDL E-14, E-124	SGP_WRHIT@ E-16, E-129
SGP_DVBSYDL E-14, E-124	SGP_WRITE@ E-16, E-129
SGP_ECMCFBS E-15, E-124	SGP_WRPRSEC E-16, E-130
SGP_ECMMSGs E-15, E-124	SGPROCACCTn D-67
SGP_ECMNSPC E-15, E-124	SGREADNTVLn D-14, D-67
SGP_ECMPGMS E-15, E-125	SGRETRYLIMn D-15, D-67
SGP_EXITBBS D-13, D-61	SGSPACHLDRn D-15, D-67
SGP_EXITLIB D-13, D-62, D-66	SGSUBTASKSn D-15, D-68
SGP_FSCBYRD E-15, E-125	SGWRITNTVLn D-15, D-68
SGP_FSCPERC E-15, E-125	shared parmlib environment 3-10
SGP_FSUPERC E-15, E-125	SHUTDOWN command 2-18
SGP_IOPRSEC E-15, E-125	SIZE E-16, E-130
SGP_IOSQTIM E-15, E-126	SIZEISPRIM D-15, D-40, D-68
SGP_LCU@BSY E-126	SKIP D-15, D-69
SGP_LCU@BUSY E-15	SMCALsxx 3-4, 8-2
SGP_LCUID E-15, E-126	parameter for calendar settings D-104

---

SMCRITxx 3-4	SMSPOL D-103
SMDIAGxx 3-4	SMSPOLxx 3-5
defining diagnostic parameters D-112	SMSPPOOL D-16, E-16, E-131
SMDIAGxx subparameters	SMSPPOOL_EXT E-16, E-132
ABEND D-112	SMVARSxx 3-5, 7-2, D-106, D-113, D-116
DEBUG D-112	Softworks 2-13
DUMP D-112	SOLUTION E-16, E-132
IGNORE D-112	SORT E-16
MODTRC D-112	space collector 2-9, 2-12
TRACE D-112	space reduction
SMF 3-4, 4-3, 5-3, 5-5, D-108, D-110, E-16, E-130	limitations D-41
SMF message library B-2	SPACPRIM E-16, E-133
SMF record format B-2	SPACSECA E-16, E-134
SMFID D-15, D-70	SPACSECB E-16, E-134
SMFLSTxx 3-4, 5-2	SPACSECI E-16, E-134
SMFUNCxx 3-4, D-107	SPACSECR E-16, E-135
SMMSYSxx 3-3, 3-4, D-2	SPACSWIR E-16, E-135
global parameters D-2	SPACVOLA D-84, D-102, E-16, E-136
SET keyword D-2	SPECIFIC E-16, E-136
subparameters	SPLIT E-16, E-136
CAL D-2	SQTY E-16, E-136
FUNC D-2	START command 2-18
MSGID D-2	START_ALL D-71
MSGPREF D-2	START_ALLOC 2-9, D-72
OPMHLQ D-2	START_AUTO 2-9, D-72
PASSWORD D-2	START_EHSM 2-9, D-16
POOL D-2	START_EPOOL 2-9, D-16, D-72
SMFID D-2	START_ESA D-16, D-72
usage notes D-22	START_ESMS 2-9, D-16, D-73
SMMSYSxx subparameters D-2	START_RPRT 2-9, D-73
SMPOOL	START_SGA 2-9, D-16, D-73
defining pool parameters D-99, D-103	START_SGC 2-9, D-16, D-73
SMPOOLxx 3-4, 6-2, D-99, D-103	START_SGD 2-9, D-16, D-74
parameters 6-2	START_SGP 2-9, D-16, D-74
SMPOOLxx subparameters	START_X37 2-9, D-16, D-74
POOLNAME D-99, D-103	starting and stopping MAINVIEW SRM 2-4
TYPE D-99	starting and stopping MAINVIEW SRM with
USELIMIT D-99	other products 2-13
SMRLSTxx 3-4, 3-5, 5-7	starting MAINVIEW SRM components 2-8, 2-9
SMS E-16, E-131	application collector 1-2, 2-9
SMS subpools 6-2	EasyPOOL 2-9
SMS_ALLOC D-15, D-70	EasySMS 2-9
SMS_EXTEND D-15, D-71	HSM collector 1-2, 2-9
SMSMANAGED 3-7, E-16, E-131	messages 2-10
SMSMCREN 5-7	performance collector 1-2, 2-9
SMSMF000 B-2	S SVALLOC 2-9
SMSMFCNV B-2	S SVESA 2-9
	S SVHSM 2-9

- S SVOS,SUF=00 2-9
- S SVSGA 2-9
- S SVSGC 2-9
- S SVSGD 2-9
- S SVSGP 2-9
- SG-Auto 2-9
- space alternate collector 1-2, 2-9
- space collector 1-2
- StopX37/II 2-9
- SVOS 2-9
- statements and parameters A-2
- static activate
  - activate
  - static 2-2
- STATUS command 2-18
- STEP 9-4, D-112, E-16, E-137
- STEPACCTn E-16, E-137
- STKSCR D-17, D-75
- STOGROUP E-16, E-137
- STOP command 2-20
- stopping MAINVIEW SRM 2-12
  - with other products 2-13
- stopping MAINVIEW SRM components 2-12
  - /P SVALLOC 2-12
  - /P SVESA 2-12
  - /P SVHSM 2-12
  - /P SVOS 2-12
  - /P SVSGA 2-12
  - /P SVSGC 2-12
  - /P SVSGD 2-12
  - /P SVSGP 2-12
  - application collector 2-12
  - EasyPOOL 2-12
  - EasySMS 2-12
  - HSM collector 2-12
  - messages 2-12
  - performance collector 2-12
  - SG-Auto 2-12
  - space collector 2-12
  - StopX37/II 2-12
- stopping SVOS 2-13
- STOPX37II D-17
- STOPXONLY D-17
- StorageGUARD renamed space collector and performance collector 1-2
- STORCLAS E-16, E-137
- STORGRP E-16, E-138
- STRIPCNT E-16, E-138

- STRIPTY E-16, E-138
- SUBCHAR
  - parm on SVOS 2-6
- SUBSYS
  - parm on SVOS 2-7
- subsystem character
  - modification 2-6
  - using 2-15
- SUPJSCAT 5-7
- SUPVOL E-16, E-138
- SUPVOLRF 5-7
- SVOS 2-14
  - multiple subsystem support 2-14
  - operator commands 2-21
  - parameters 2-5
- SVOS command definitions 2-16
- SVOS command syntax 2-14
- SVOS commands
  - DUMP 2-16
  - FORCE 2-16
  - SHUTDOWN 2-16
  - START 2-16
  - STAT 2-16
  - STATUS 2-16
  - STOP 2-16
- SVSGC 2-14
- switching to a cached device D-69
- syntax
  - masking A-6
  - shared parmlib environment 3-11
- syntax rules A-1
- SYSID E-16, E-139
- SYSLIB D-17, D-76
- SYSLIB2 D-17
- SYSLIB3 D-17
- sysplex
  - defining environment 3-10
  - parameters 3-11
- system configuration members 3-3

## T

- TAPE\_CA1DSN D-17, D-77
- TAPE\_CAT D-17, D-77
- TAPE\_CCTLTH D-17, D-77
- TAPE\_CHLQ D-17, D-78
- TAPE\_CPRI D-17, D-78

---

TAPE\_CSEC D-17, D-78  
TAPE\_CVOL D-18, D-79  
TAPEDEFR 5-7  
TAPEGENR D-18, D-80  
TEMPDSN E-16, E-139  
Top Secret E-91, E-140  
TRACE 4-3, 9-4, 9-5, D-81, D-108, D-111  
    options 9-10  
TRACEDD 9-10, D-18, D-81  
TRKCYL D-18, D-81, E-16, E-139  
TRKLEN D-18, D-82, E-16, E-139  
TSO D-108, D-111  
TYPE 6-3, D-99, D-101, D-112

## U

UNIT E-17, E-140  
UPGRADE  
    in VSAMDEF E-145  
USECAT D-18, D-82  
USECPOOL E-17, E-140  
USELIMIT 6-3, D-99, D-102, D-112  
USER E-17, E-140  
USERVARS 4-5  
    example 4-6  
    function 4-5  
    parameters 4-5  
USEVOL E-17, E-141  
USRCn 4-6, E-17, E-141  
USRCy E-17  
USRNy 4-6, E-141

## V

VALUE 7-2, D-106, D-107  
VAR D-18, D-83  
VARIABLE 7-2, D-106, D-107  
variables 3-10  
    examples 7-4  
    name masks A-8  
    parameters D-106  
    usage 7-4  
VASMLIMWARN D-84  
VASMPRIM D-84  
VCOMPLLQ E-17, E-142  
version number

SVOS 2-19  
VFORCE E-17, E-142  
view  
    displaying  
        from menus 1-4  
        menu-driven 1-4  
VIO E-17, E-142  
VOL 3-6, 3-7, 6-3, D-99, D-102, D-103, D-104,  
    E-17, E-142  
VOLSEL D-102, E-17, E-143  
VOLSER E-17, E-144  
volume selection D-81, D-82  
volume switch  
    cached devices D-29  
    device characteristics D-29  
    dualcopy D-29  
    fastwrite D-29  
    same device characteristics D-69  
    shared devices D-29  
VSAM D-84  
    data set cluster definition E-145  
    data set component type E-144  
    RECORD, record organization parameter  
        E-93  
VSAMCOMP E-17, E-144  
VSAMDEF E-17, E-145  
VSAMJCL D-18, D-83  
VSAMLIMWARN D-19  
VSAMPRIM D-19  
VSAMSEP E-17, E-145  
VSAMZSEC D-19, D-84  
VSCAN\_AGER D-19  
VSCAN\_AGER1= D-85  
VSCAN\_AGER2 D-85  
VSCAN\_AGER3 D-85  
VSCAN\_AGER4 D-86  
VSCAN\_AGER5 D-86  
VSCAN\_AGER6 D-86  
VSCAN\_AGER7 D-87  
VSCAN\_AGER8 D-87  
VSCAN\_AGER9 D-87  
VSCAN\_MNTSK D-21, D-94  
VSCAN\_MXTSK D-21, D-94  
VSCAN\_OINDX D-21, D-94  
VSCAN\_OPRI D-21, D-95  
VSCAN\_OSEC D-21, D-95  
VSCAN\_OUNIT D-21, D-95  
VSCAN\_OVOL D-21, D-95

---

VSCAN\_PCTR D-19  
VSCAN\_PCTR1 D-88  
VSCAN\_PCTR2 D-88  
VSCAN\_PCTR3 D-88  
VSCAN\_PCTR4 D-89  
VSCAN\_PCTR5 D-89  
VSCAN\_PCTR6 D-89  
VSCAN\_PCTR7 D-90  
VSCAN\_PCTR8 D-90  
VSCAN\_PCTR9 D-90  
VSCAN\_PCTRA D-91  
VSCAN\_SIZR D-20  
VSCAN\_SIZR1 D-91  
VSCAN\_SIZR2 D-91  
VSCAN\_SIZR3 D-91  
VSCAN\_SIZR4 D-92  
VSCAN\_SIZR5 D-92  
VSCAN\_SIZR6 D-92  
VSCAN\_SIZR7 D-93  
VSCAN\_SIZR8 D-93  
VSCAN\_SIZR9 D-93  
VSCAN\_TPRI D-21, D-96  
VSCAN\_TSEC D-21, D-96  
VSCAN\_TUNIT D-21, D-96  
VSCAN\_TVOL D-21, D-97

## **W**

WTODC D-22, D-97  
WTORC D-22, D-97

## **X**

X37POOL D-22, D-97  
XMODE E-17, E-145

## **Y**

YEAR 8-2, D-104, D-105



# END USER LICENSE AGREEMENT NOTICE

BY OPENING THE PACKAGE, INSTALLING, PRESSING "AGREE" OR "YES" OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT INSTALL OR USE THE PRODUCT, PROMPTLY RETURN THE PRODUCT TO BMC OR YOUR BMC RESELLER, AND IF YOU ACQUIRED THE LICENSE WITHIN 30 DAYS OF THE DATE OF YOUR ORDER CONTACT BMC OR YOUR BMC RESELLER FOR A REFUND OF LICENSE FEES PAID. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.

This Agreement ("**Agreement**") is between the entity or individual entering into this Agreement ("You") and BMC Software Distribution, Inc., a Delaware corporation located at 2101 CityWest Blvd., Houston, Texas, 77042, USA or its affiliated local licensing entity ("BMC"). "You" includes you and your Affiliates. "Affiliate" is defined as an entity which controls, is controlled by or shares common control with a party. THIS AGREEMENT WILL APPLY TO THE PRODUCT, UNLESS (1) YOU AGREED TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT, IN WHICH CASE THAT WEB BASED LICENSE AGREEMENT GOVERNS THE USE OF THE PRODUCT, OR (2) IF YOU DID NOT AGREE TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT AND YOU HAVE A WRITTEN LICENSE AGREEMENT WITH BMC, THEN THAT WRITTEN AGREEMENT GOVERNS THE USE OF THE PRODUCT. THE ELECTRONIC AGREEMENT PROVIDED WITH THE PRODUCT AS PART OF THE INSTALLATION OF THE PRODUCT WILL NOT APPLY. In addition to the restrictions imposed under this Agreement, any other usage restrictions contained in the Product installation instructions or release notes shall apply to Your use of the Product.

**PRODUCT AND CAPACITY.** "**Software**" means the object code version of the computer programs provided, via delivery or electronic transmission, to You. Software includes computer files, enhancements, maintenance modifications, upgrades, updates, bug fixes, and error corrections.

"**Documentation**" means all written or graphical material provided by BMC in any medium, including any technical specifications, relating to the functionality or operation of the Software.

"**Product**" means the Software and Documentation.

"**License Capacity**" means the licensed capacity for the Software with the pricing and other license defining terms, including capacity restrictions, such as tier limit, total allowed users, gigabyte limit, quantity of Software, and/or other capacity limitations regarding the Software. For licenses based on the power of a computer, You agree to use BMC's current computer classification scheme, which is available at <http://www.bmc.com> or can be provided to You upon request.

**ACCEPTANCE.** The Product is deemed accepted by You, on the date that You received the Product from BMC.

**LICENSE.** Subject to the terms of this Agreement, as well as Your payment of applicable fees, BMC grants You a non-exclusive, non-transferable, perpetual (unless a term license is provided on an order) license for each copy of the Software, up to the License Capacity, to do the following:

- (a) install the Software on Your owned or leased hardware located at a facility owned or controlled by You in the country where You acquired the license;
- (b) operate the Software solely for processing Your own data in Your business operations; and
- (c) make one copy of the Software for backup and archival purposes only (collectively a "**License**").

If the Software is designed by BMC to permit you to modify such Software, then you agree to only use such modifications or new software programs for Your internal purposes or otherwise consistent with the License. BMC grants You a license to use the Documentation solely for Your internal use in Your operations.

**LICENSE UPGRADES.** You may expand the scope of the License Capacity only pursuant to a separate agreement with BMC for such expanded usage and Your payment of applicable fees. There is no additional warranty period or free support period for license upgrades.

**RESTRICTIONS:** You agree to **NOT**:

- (a) disassemble, reverse engineer, decompile or otherwise attempt to derive any Software from executable code;
- (b) distribute or provide the Software to any third party (including without limitation, use in a service bureau, outsourcing environment, or processing the data of third parties, or for rental, lease, or sublicense); or
- (c) provide a third party with the results of any functional evaluation or benchmarking or performance tests, without BMC's prior written approval, unless prohibited by local law.

**TRIAL LICENSE.** If, as part of the ordering process, the Product is provided on a trial basis, then these terms apply: (i) this license consists solely of a non-exclusive, non-transferable evaluation license to operate the Software for the period of time specified from BMC or, if not specified, a 30 day time period ("**Trial Period**") only for evaluating whether You desire to acquire a capacity-based license to the Product for a fee; and (ii) Your use of the Product is on an AS IS basis without any warranty, and **BMC, ITS AFFILIATES AND RESELLERS, AND LICENSORS DISCLAIM ANY AND ALL WARRANTIES (INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT) AND HAVE NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE ("Trial License").** BMC may terminate for its convenience a Trial License upon notice to You. When the Trial Period ends, Your right to use this Product automatically expires. If You want to continue Your use of the Product beyond the Trial Period, contact BMC to acquire a capacity-based license to the Product for a fee.

**TERMINATION.** This Agreement shall immediately terminate if You breach any of its terms. Upon termination, for any reason, You must uninstall the Software, and either certify the destruction of the Product or return it to BMC.

**OWNERSHIP OF THE PRODUCT.** BMC or its Affiliates or licensors retain all right, title and interest to and in the BMC Product and all intellectual property, informational, industrial property and proprietary rights therein. BMC neither grants nor otherwise transfers any rights of ownership in the BMC Product to You. BMC Products are protected by applicable copyright, trade secret, and industrial and intellectual property laws. BMC reserves any rights not expressly granted to You herein.

**CONFIDENTIAL AND PROPRIETARY INFORMATION.** The BMC Products are and contain valuable confidential information of BMC ("Confidential Information"). Confidential Information means non-public technical and non-technical information relating to the BMC Products and Support, including, without limitation, trade secret and proprietary information, and the structure and organization of the Software. You may not disclose the Confidential Information to third parties. You agree to use all reasonable efforts to prevent the unauthorized use, copying, publication or dissemination of the Product.

**WARRANTY.** Except for a Trial License, BMC warrants that the Software will perform in substantial accordance with the Documentation for a period of one year from the date of the order. This warranty shall not apply to any problems caused by software or hardware not supplied by BMC or to any misuse of the Software.

**EXCLUSIVE REMEDY.** BMC's entire liability, and Your exclusive remedy, for any defect in the Software during the warranty period or breach of the warranty above shall be limited to the following: BMC shall use reasonable efforts to remedy defects covered by the warranty or replace the defective Software within a reasonable period of time, or if BMC cannot remedy or replace such defective copy of the Software, then BMC shall refund the amount paid by You for the License for that Software. BMC's obligations in this section are conditioned upon Your providing BMC prompt access to the affected Software and full cooperation in resolving the claim.

**DISCLAIMER. EXCEPT FOR THE EXPRESS WARRANTIES ABOVE, THE PRODUCT IS PROVIDED "AS IS." BMC, ITS AFFILIATES AND LICENSORS SPECIFICALLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. BMC DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS CAN BE CORRECTED.**

**DISCLAIMER OF DAMAGES. IN NO EVENT IS BMC, ITS AFFILIATES OR LICENSORS LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RELATING TO OR ARISING OUT OF THIS AGREEMENT, SUPPORT, AND/OR THE PRODUCT (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST COMPUTER USAGE TIME, AND DAMAGE OR LOSS OF USE OF DATA), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IRRESPECTIVE OF ANY NEGLIGENCE OF BMC OR WHETHER SUCH DAMAGES RESULT FROM A CLAIM ARISING UNDER TORT OR CONTRACT LAW.**

**LIMITS ON LIABILITY. BMC'S AGGREGATE LIABILITY FOR DAMAGES IS LIMITED TO THE AMOUNT PAID BY YOU FOR THE LICENSE TO THE PRODUCT.**

**SUPPORT.** If Your order includes support for the Software, then BMC agrees to provide support (24 hours a day/7 days a week) ("Support"). You will be automatically re-enrolled in Support on an annual basis unless BMC receives notice of termination from You as provided below. There is a free support period during the one year warranty period.

(a) **Support Terms.** BMC agrees to make commercially reasonable efforts to provide the following Support: (i) For malfunctions of supported versions of the Software, BMC provides bug fixes, patches or workarounds in order to cause that copy of the Software to operate in substantial conformity with its then-current operating specifications; and (ii) BMC provides new releases or versions, so long as such new releases or versions are furnished by BMC to all other enrolled Support customers without additional charge. BMC may refuse to provide Support for any versions or releases of the Software other than the most recent version or release of such Software made available by BMC. Either party may terminate Your enrollment in Support upon providing notice to the other at least 30 days prior to the next applicable Support anniversary date. If You re-enroll in Support, BMC may charge You a reinstatement fee of 1.5 times what You would have paid if You were enrolled in Support during that time period.

(b) **Fees.** The annual fee for Support is 20% of the Software's list price less the applicable discount or a flat capacity based annual fee. BMC may change its prices for the Software and/or Support upon at least 30 days notice prior to Your support anniversary date.

**VERIFICATION.** If requested by BMC, You agree to deliver to BMC periodic written reports, whether generated manually or electronically, detailing Your use of the Software in accordance with this Agreement, including, without limitation, the License Capacity. BMC may, at its expense, audit Your use of the Software to confirm Your compliance with the Agreement. If an audit reveals that You have underpaid fees, You agree to pay such underpaid fees. If the underpaid fees exceed 5% of the fees paid, then You agree to also pay BMC's reasonable costs of conducting the audit.

**EXPORT CONTROLS.** You agree not to import, export, re-export, or transfer, directly or indirectly, any part of the Product or any underlying information or technology except in full compliance with all United States, foreign and other applicable laws and regulations.

**GOVERNING LAW.** This Agreement is governed by the substantive laws in force, without regard to conflict of laws principles: (a) in the State of New York, if you acquired the License in the United States, Puerto Rico, or any country in Central or South America; (b) in the Province of Ontario, if you acquired the License in Canada (subsections (a) and (b) collectively referred to as the "**Americas Region**"); (c) in Singapore, if you acquired the License in Japan, South Korea, Peoples Republic of China, Special Administrative Region of Hong Kong, Republic of China, Philippines, Indonesia, Malaysia, Singapore, India, Australia, New Zealand, or Thailand (collectively, "**Asia Pacific Region**"); or (d) in the Netherlands, if you acquired the License in any other country not described above. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed in its entirety.

**ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL OR INTERNATIONAL, AS APPLICABLE, RULES OF THE AMERICAN ARBITRATION ASSOCIATION. IF THE DISPUTE IS INITIATED IN A COUNTRY IN THE ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN SINGAPORE, SINGAPORE UNDER THE CURRENT UNCITRAL ARBITRATION RULES. IF THE DISPUTE IS INITIATED IN A COUNTRY OUTSIDE OF THE AMERICAS REGION OR ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN AMSTERDAM, NETHERLANDS UNDER THE CURRENT UNCITRAL ARBITRATION RULES. THE COSTS OF THE ARBITRATION SHALL BE BORNE EQUALLY PENDING THE ARBITRATOR'S AWARD. THE AWARD RENDERED SHALL BE FINAL AND BINDING UPON THE PARTIES AND SHALL NOT BE SUBJECT TO APPEAL TO ANY COURT, AND MAY BE ENFORCED IN ANY COURT OF COMPETENT JURISDICTION. NOTHING IN THIS AGREEMENT SHALL BE DEEMED AS PREVENTING EITHER PARTY FROM SEEKING INJUNCTIVE RELIEF FROM ANY COURT HAVING JURISDICTION OVER THE PARTIES AND THE SUBJECT MATTER OF THE DISPUTE AS NECESSARY TO PROTECT EITHER PARTY'S CONFIDENTIAL INFORMATION, OWNERSHIP, OR ANY OTHER**



**PROPRIETARY RIGHTS. ALL ARBITRATION PROCEEDINGS SHALL BE CONDUCTED IN CONFIDENCE, AND THE PARTY PREVAILING IN ARBITRATION SHALL BE ENTITLED TO RECOVER ITS REASONABLE ATTORNEYS' FEES AND NECESSARY COSTS INCURRED RELATED THERETO FROM THE OTHER PARTY.**

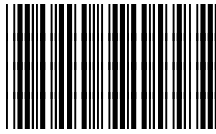
**U.S. GOVERNMENT RESTRICTED RIGHTS.** The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("**FAR**") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("**DOD**"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202 of the DOD FAR Supplement and its successors.

**MISCELLANEOUS TERMS.** You agree to pay BMC all amounts owed no later than 30 days from the date of the applicable invoice, unless otherwise provided on the order for the License to the Products. You will pay, or reimburse BMC, for taxes of any kind, including sales, use, duty, tariffs, customs, withholding, property, value-added (VAT), and other similar federal, state or local taxes (other than taxes based on BMC's net income) imposed in connection with the Product and/or the Support. This Agreement constitutes the entire agreement between You and BMC and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both BMC and You. You may not assign or transfer this Agreement or a License to a third party without BMC's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

SW EULA Int 030102



## Notes



\*25768\*